21554 PCI-to-PCI Bridge for Embedded Applications

Hardware Reference Manual

September 1998

Order Number: 278091-001

Contents

Figures

Tables

Preface 1

This manual provides a detailed functional and register description of the 21554 PCI-to-PCI bridge device for embedded applications. For pinouts, mechanical specifications, and electrical specifications for this device, please refer to the *21554 PCI-to-PCI Bridge for Embedded Applications Data Sheet*.

1.1 Manual Organization

This manual contains the following chapters and an appendix:

Chapter 2, “Introduction”, provides an overview of the 21554 functionality and architecture.

Chapter 3, “Signal Pins”, describes the signal pins grouped by function.

Chapter 4, “PCI Bus Operation”, PCI transactions, transaction forwarding, and transaction termination.

Chapter 5, “Address Decoding”, contains details about how addresses are decoded.

Chapter 6, “Configuration Accesses”, explains how 21554 responds to Type 0 configuration accesses.

Chapter 7, “Configuration Space Registers”, describes the 21554 configuration registers.

Chapter 8, “Control and Status Registers”, describes the 21554 CSRs.

Chapter 9, “I2O Support”, explains how the 21554 implements an I_2O messaging unit.

Chapter 10, “Interrupt and Scratchpad Registers”, describes interrupt support and scratchpad registers.

Chapter 11, “Parallel ROM Interface”, describes the 21554 parallel ROM interface.

Chapter 12, “Serial ROM Interface”, describes the 21554 serial ROM interface.

Chapter 13, “Arbitration”, explains how 21554 implements primary and secondary PCI bus arbitration.

Chapter 14, “Error Handling”, describes parity error responses and system error reporting.

Chapter 15, “Clocking”, describes clocking support in the 21554.

Chapter 16, “Initialization Requirements”, reset operation and initialization requirements for the 21554.

Chapter 17, “Diagnostic Mechanisms”, explains the implementation of the 21554’s JTAG test port.

Chapter 18, “VPD Support”, 21554 Vital Product Data (VPD) support through serial ROM interface.

Chapter 19, “Special Applications”, describes primary and secondary bus VGA support.

Chapter 20, “Serial ROM and Register Reset Summary”, SROM preload sequence and register reset and access.

Appendix A, “Support, Products, and Documentation”, technical support and ordering information.

1.2 Conventions and Terminology

This section describes the terminology and conventions used in this manual.

1.2.1 Caution

Cautions provide information to prevent damage to equipment or loss of data.

1.2.2 Data Units

This manual uses the following data-unit terminology.

Term	Words	Bytes	Bits
Byte	½	1	8
Word	1	2	16
Dword	2	4	32
Quadword	4	8	64

1.2.3 Note

Notes emphasize particularly important information.

1.2.4 Numbering

All numbers are decimal or hexadecimal unless otherwise indicated. In cases of ambiguity, a subscript indicates the radix of nondecimal numbers. For example, 19 is decimal, but 19h and 19A are hexadecimal.

1.2.5 Signal Names

Signal names are printed in lowercase type.

Signal names indicate whether a signal is low-asserted (the signal is active, or asserted, when it is at a low voltage level) or high-asserted (the signal is asserted when it is at a high voltage level). The names of low-asserted signals carry the suffix _l; the names of high-asserted signals have no suffix. For example, p_idsel is a high-asserted signal, and p_frame_l is a low-asserted signal.

The prefix p_ denotes a primary bus signal; the prefix s_ denotes a secondary bus signal. For example, p_ad is the primary interface address/data bus, and s_ad is the secondary interface address/data bus.

1.2.6 SIGNAME#

PCI signals that can be on either the primary interface or the secondary interface are printed in uppercase, normal type. The names of low-asserted signals are followed by #. For example, “asserting FRAME#” can refer to the assertion of the p_frame_l signal if the transaction is occurring on the primary bus, or the assertion of the s_frame_l signal if the transaction is occurring on the secondary bus.

Introduction 2

Intel's 21554 is a PCI peripheral device that performs PCI bridging functions for embedded and intelligent I/O applications. The 21554 has a 64-bit primary interface, a 64-bit secondary interface, and 33-MHz capability.

The 21554 is a "nontransparent" PCI-to-PCI bridge that acts as a gateway to an intelligent subsystem. It allows a local processor to independently configure and control the local subsystem. The 21554 implements an I_2O message unit that enables any local processor to function as an intelligent I/O processor (IOP) in an I_2O-capable system. Because the 21554 is architecture independent, it works with any host and local processors that support a PCI bus. This architecture independence enables vendors to leverage existing investments while moving products to PCI technology.

Unlike a transparent PCI-to-PCI bridge, the 21554 is specifically designed to bridge between two processor domains. The processor domain on the primary interface of the 21554 is also referred to as the host domain, and its processor is the host processor. The secondary bus interfaces to the local domain and the local processor. Special features include support of independent primary and secondary PCI clocks, independent primary and secondary address spaces, and address translation between the primary (host) and secondary (local) domains.

The 21554 enables add-in card vendors to present to the host system a higher level of abstraction than is possible with a transparent PCI-to-PCI bridge. The 21554 uses a Type 0 configuration header, which presents the entire subsystem as a single "device" to the host processor. This allows loading of a single device driver for the entire subsystem, and independent local processor initialization and control of the subsystem devices. Because the 21554 uses a Type 0 configuration header, it does not require hierarchical PCI-to-PCI bridge configuration code.

The 21554 forwards transactions between the primary and secondary PCI buses as does a transparent PCI-to-PCI bridge. In contrast to a transparent PCI-to-PCI bridge, however, the 21554 can translate the address of a forwarded transaction from a system address to a local address, or vice versa. This mechanism allows the 21554 to hide subsystem resources from the host processor and to resolve any resource conflicts that may exist between the host and local subsystems.

The 21554 operates at 3.3 V, but is also 5.0-V I/O tolerant. Adapter cards designed using the 21554 can be keyed as universal, thus permitting use in either a 5-V or 3-V slot.

2.1 Features

The 21554 also supports the following features:

PCI Interfaces

- Full compliance with the *PCI Local Bus Specification, Revision 2.1*, plus:
  - — Vital Product Data (VPD) support
  - — Compact PCI Distributed Hot-Swap support
- 3.3-V operation with 5.0-V tolerant I/O
- Selectable asynchronous or synchronous primary and secondary interface clocks

- Concurrent primary and secondary bus operation

PCI Power Management

- Fully compliant with the *Advanced Configuration Power Interface* (ACPI) specification
- Fully compliant with the *PCI Bus Power Management Specification*

Buffer Architecture

- Queuing of multiple transactions in either direction
- 256 bytes of posted write (data and address) buffering in each direction:
  - For forwarding of memory write (MW) and memory write and invalidate (MWI) transactions
- 256 bytes of read data buffering in each direction
- Four delayed transaction entries in each direction:
  - For forwarding of I/O write and all read transactions
  - For configuration and I/O transaction generation
- Two dedicated I_2O delayed transaction entries

Configuration Registers and CSRs

- Two sets of standard PCI configuration registers corresponding to the primary and secondary interface; each set is accessible from either the primary or secondary interface
- Four 32-bit base address configuration registers mapping the 21554 control and status registers (CSRs):
  - One for memory mapping of registers from the primary interface
  - One for I/O mapping of registers from the primary interface
  - One for memory mapping of registers from the secondary interface
  - One for I/O mapping of registers from the secondary interface

Transaction Forwarding

- Four primary interface base address configuration registers for downstream forwarding, with size and prefetchability programmable for all four address ranges:
  - One programmable for forwarding of either I/O or memory transactions
  - Three for forwarding of memory transactions:

    *One shared with primary CSR memory mapping.

    *One is configurable as a 64-bit address base address register (BAR) for downstream dual-address cycles (DACs).
- Direct offset address translation for downstream memory and I/O transactions
- Three secondary interface address configuration registers specifying local address ranges for upstream forwarding, with size and prefetchability programmable for all three address ranges:
  - One programmable for forwarding of either I/O or memory transactions using direct offset address translation
  - One for forwarding memory transactions using direct offset address translation

  - One for forwarding memory transactions using lookup table (LUT) based address translation
- Inverse decoding above the 4GB address boundary for upstream DACs
- Ability to generate Type 0 and Type 1 configuration commands on the primary or secondary interface via configuration or I/O CSR accesses
- Ability to generate I/O commands on the primary or secondary interface via I/O CSR accesses

Intelligent I/O Support

- I_2O message unit
  - Standard I_2O registers at 40h and 44h, with queue-not-empty interrupts
  - Queue size selectable between 256 and 32K entries, by powers of 2
  - Hardware support for all primary (host-side) queue pointers
  - Hardware support for queue-not-empty detection
- Doorbell registers for software generation of primary and secondary bus interrupts, 16 bits per interface
- Eight Dwords of scratchpad registers

ROM Interfaces

- Parallel flash ROM interface with primary bus expansion ROM base address register
  - Read- and write-accessible by CSR access
  - Programmable expansion ROM window size from 4KB to 16MB
  - Programmable access times
- Serial ROM interface
  - Read and write accessible through CSR access
  - Enables preloading of selected configuration register fields to application-specific values
    * Subsystem ID, subsystem vendor ID, and class code
    * Numbers, sizes, and types of base address registers
    * MIN_GNT and MAX_LAT for both interfaces
    * Device-specific control bits

Miscellaneous Functions

- Secondary bus arbiter support for up to nine external devices (in addition to the 21554)
  - Programmable two-level, rotating-priority arbiter
  - Hardware disable control for internal arbiter to allow use of an external arbiter
- Secondary bus clock output for synchronous operation
  - May be buffered externally to support any number of secondary bus devices
    * Buffered version is fed back to the 21554 secondary clock input
  - Hardware and software disable control
- Hardware enable for secondary bus central functions

— Driving s_ad[31:0], s_cbe_l[3:0], and s_par during reset

— Asserting s_req64_l during reset

- IEEE Standard 1149.1 boundary-scan JTAG interface

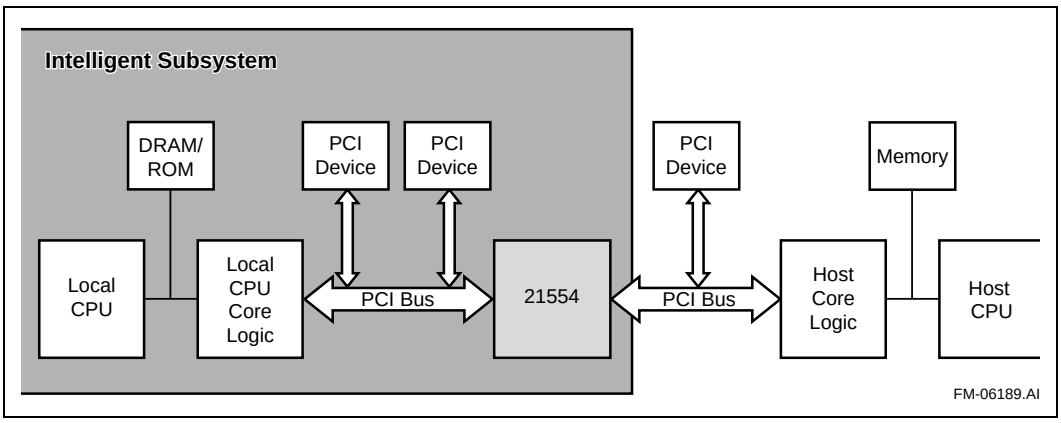
2.2 Comparing 21554 and Standard PCI-to-PCI Bridge

The 21554 is functionally similar to a standard PCI-to-PCI bridge (PPB) in that both provide a connection path between devices attached to two independent PCI buses. A 21554 and a PPB allow the electrical loading of devices on one PCI bus to be isolated from the other bus while permitting concurrent operation on both buses. Because the *PCI Local Bus Specification* restricts PCI option cards to a single electrical load, the ability of PPBs and the 21554 to spawn PCI buses enables the design of multidevice PCI option cards. The key difference between a PPB and the 21554 is that the presence of a PPB in a connection path between the host processor and a device is transparent to devices and device drivers, while the presence of the 21554 is not. This difference enables the 21554 to provide features that better support the use of intelligent controllers in the subsystem.

It was a primary goal of the PCI-to-PCI bridge architecture that a PPB be transparent to devices and device drivers. For example, no changes are needed to a device driver when a PCI peripheral is located behind a PPB. Once configured during system initialization, a PPB operates without the aid of a device driver. A PPB does not require a device driver of its own since it does not have any resources that must be managed by software during run-time. This requirement for transparency forced the usage of a flat addressing model across PCI-to-PCI bridges. This means that a given physical address exists at only one location in the PCI bus hierarchy and that this location may be accessed by any device attached at any point in the PCI bus hierarchy. As a consequence, it is not possible for a PPB to isolate devices or address ranges from access by devices on the opposite interface of a PPB. The PPB architecture assumes that the resources of any device in a PCI system are configured and managed by the host processor.

However, there are applications where the transparency of a PCI-to-PCI bridge is not desired. For example, Figure 2-1 shows a hypothetical PCI add-in card used for an intelligent subsystem application.

Figure 2-1. 21554 Intelligent Controller Application



Assume that the local processor on the add-in card is used to manage the resources of the devices attached to the add-in card's local PCI bus. Assume also that it is desirable to restrict access to these same resources from other PCI bus masters in the system and from the host processor. In

addition, there is a need to resolve address conflicts that may exist between the host system and the local processor. The nontransparency of the 21554 is perfectly suited to this kind of configuration, where a transparent PCI-to-PCI bridge would be problematic.

Because the 21554 is not transparent, the device driver for the add-in card must be aware of the presence of the 21554 and manage its resources appropriately. The 21554 allows the entire subsystem to appear as a single virtual device to the host. This enables configuration software to identify the appropriate driver for the subsystem.

With a transparent PCI-to-PCI bridge, a driver does not need to know about the presence of the bridge and manage its resources. The subsystem appears to the host system as individual PCI devices on a secondary PCI bus, not as a single virtual device.

Table 2-1 shows a comparison between a 21554 and a standard transparent PCI-to-PCI bridge.

Table 2-1. 21554 and PPB Feature Comparison

Feature	21554	PCI-to-PCI Bridge
Transaction forwarding	Adheres to PPB ordering rules. Uses posted writes and delayed transactions. Adheres to PPB transaction error and parity error guidelines, although some errors may be reported differently.	Adheres to PPB ordering rules. Uses posted writes and delayed transactions. Adheres to PPB transaction error and parity error guidelines.
Address decoding	Base address registers are used to define independent downstream and upstream forwarding windows. Inverse decoding is only used for upstream transactions above the 4GB boundary.	PPB base and limit address registers are used to define downstream forwarding windows. Inverse decoding for all I/O and memory upstream forwarding.
Address translation	Supported for both memory and I/O transactions.	None. Flat address model is assumed.
Configuration	Downstream devices are not visible to host. Does not require hierarchical configuration code (Type 0 configuration header). Does not respond to Type 1 configuration transactions. Supports configuration access from the secondary bus. Implements separate set of configuration registers for the secondary interface.	Downstream devices are visible to host. Requires hierarchical configuration code (Type 1 configuration header). Forwards and converts Type 1 configuration transactions. Does not support configuration access from the secondary bus. Same set of configuration registers is used to control both primary and secondary interfaces.
Run-time resources	Includes features such as doorbell interrupts, I_2O message unit, and so on, that must be managed by the device driver.	Typically has only configuration registers; no device driver is required.
Clocks	Generates secondary bus clock output. Asynchronous secondary clock input is also supported.	Generates one or more secondary bus clock outputs.
Secondary bus central functions	Implements secondary bus arbiter. This function can be disabled. Drives secondary bus AD, C/BE#, and PAR during reset. This function can be disabled.	Implements secondary bus arbiter. This function can be disabled. Drives secondary bus AD, C/BE#, and PAR during reset.

2.3 Architectural Overview

The 21554 consists of the following function blocks:

Data Buffers

Data buffers include the buffers along with the associated data path control logic. Delayed transaction buffers contain the compare functionality for completing delayed transactions. The blocks also contain the watchdog timers associated with the buffers. The data buffers are as follows:

- Four-entry downstream delayed transaction buffer
- Four-entry upstream delayed transaction buffer
- 256-byte downstream posted write buffer
- 256-byte upstream posted write buffer
- 256-byte downstream read data buffer
- 256-byte upstream read data buffer
- Two downstream I_2O delayed transaction entries

Registers

The following register blocks also contain address decode and translation logic, I_2O message unit, and interrupt control logic:

- Primary interface header Type 0 configuration registers
- Secondary interface header Type 0 configuration registers
- Device-specific configuration registers
- Memory and I/O mapped control and status registers

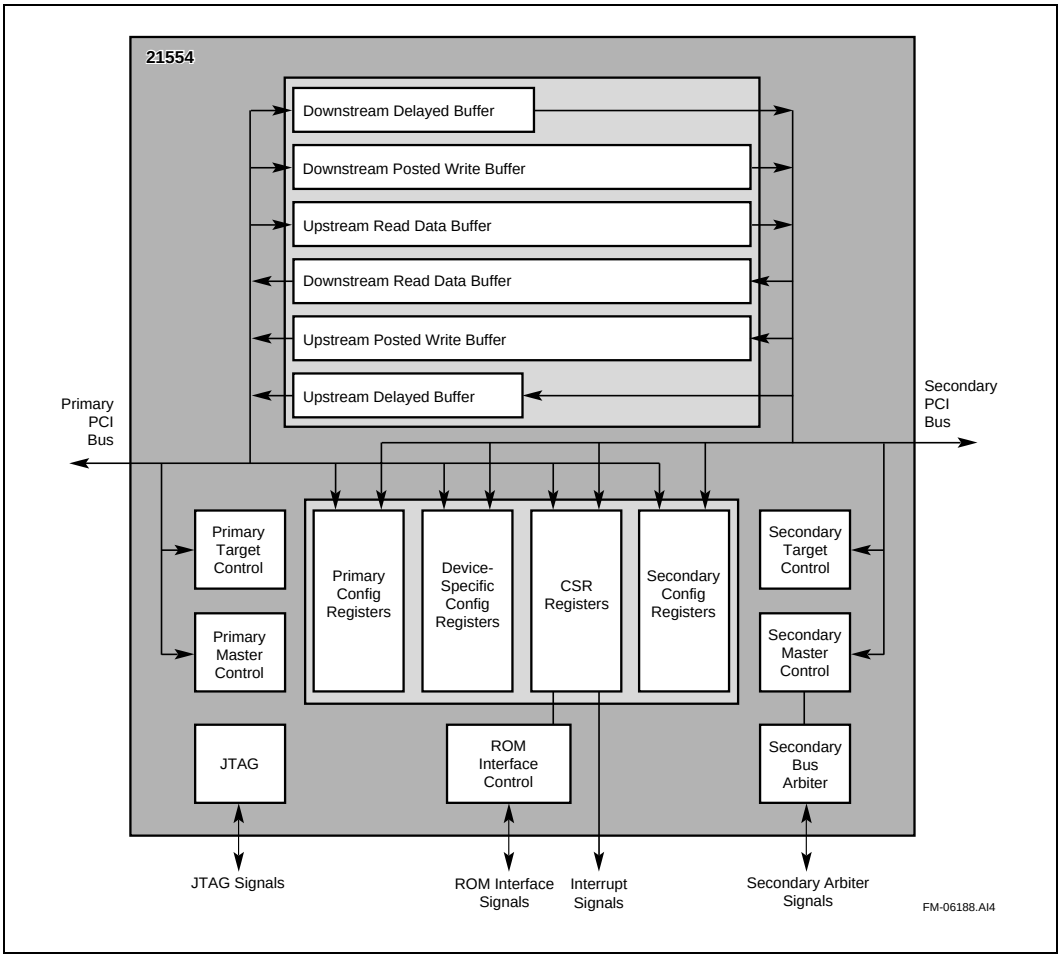
Control Logic

The 21554 has the following control logic:

- Primary PCI target control logic
- Primary PCI master control logic
- Secondary PCI target control logic
- Secondary PCI master control logic
- ROM interface control logic for both serial and parallel ROM connections (interfaces between the ROM registers and ROM signals)
- Secondary PCI bus arbiter interface to secondary bus device request and grant lines, as well as the 21554 secondary master control logic
- JTAG control logic

Figure 2-2 shows the 21554 microarchitecture.

Figure 2-2. 21554 Microarchitecture



Signal Pins 3

This chapter provides detailed descriptions of the 21554 signal pins, grouped by function. Table 3-1 describes these signal pin functional groups.

Table 3-1. Signal Pin Functional Groups

Function	Description
Primary PCI bus interface signal pins	All PCI pins required by the *PCI Local Bus Specification, Revision 2.1.*
Primary PCI bus interface 64-bit extension signal pins	All PCI 64-bit extension pins required by the *PCI Local Bus Specification, Revision 2.1.*
Secondary PCI bus interface signal pins	All PCI pins required by the *PCI Local Bus Specification, Revision 2.1.*
Secondary PCI bus interface 64-bit extension signal pins	All PCI 64-bit extension pins required by the *PCI Local Bus Specification, Revision 2.1.*
Secondary PCI bus arbitration signal pins	Nine request/grant pairs of pins for the secondary PCI bus.
Clock signal pins	Two clock inputs (one for each PCI interface). One secondary bus clock output.
Power management, hot swap, and reset signal pins	Power management and hot-swap status and events. A primary interface reset input. A secondary interface reset output.
ROM interface signal pins	8-bit multiplexed address/data bus plus control for parallel and serial ROM connection.
Miscellaneous signal pins	Two input voltage signaling level pins.
Diagnostic signal pins	IEEE Standard 1149.1 boundary-scan JTAG interface. Scan enable.

Table 3-2 defines the signal type abbreviations used in the signal tables.

Table 3-2. Signal Type Abbreviations

Signal Type	Description
I	Standard input only.
O	Standard output only.
TS	Tristate bidirectional.
STS	Sustained tristate. Active low signal must be pulled high for one clock cycle when deasserting.
OD	Standard open drain.

Note: The _l signal name suffix indicates that the signal is asserted when it is at a low voltage level and corresponds to the # suffix in the *PCI Local Bus Specification, Revision 2.1*. If this suffix is not present, the signal is asserted when it is at a high voltage level.

3.1 Primary PCI Bus Interface Signals

Table 3-3 describes the primary PCI bus interface signals.

Table 3-3. Primary PCI Bus Interface Signals (Sheet 1 of 3)

Signal Name	Type	Description
p_ad[31:0]	TS	Primary PCI interface address/data. These signals are a 32-bit multiplexed address and data bus. During the address phase or phases of a transaction, the initiator drives a physical address on p_ad[31:0]. During the data phases of a transaction, the initiator drives write data, or the target drives read data, on p_ad[31:0]. When the primary PCI bus is idle, the 21554 drives p_ad to a valid logic level when p_gnt_l is asserted.
p_cbe_l[3:0]	TS	Primary PCI interface command/byte enables. These signals are a multiplexed command field and byte enable field. During the address phase or phases of a transaction, the initiator drives the transaction type on p_cbe_l[3:0]. When there are two address phases, the first address phase carries the dual-address command and the second address phase carries the transaction type. For both read and write transactions, the initiator drives byte enables on p_cbe_l[3:0] during the data phases. When the primary PCI bus is idle, the 21554 drives p_cbe_l to a valid logic level when p_gnt_l is asserted.
p_devsel_l	STS	Primary PCI interface DEVSEL#. Signal p_devsel_l is asserted by the target, indicating that the device is responding to the transaction. As a target, the 21554 decodes the address of a transaction initiated on the primary bus to determine whether to assert p_devsel_l. As an initiator of a transaction on the primary bus, the 21554 looks for the assertion of p_devsel_l within five clock cycles of p_frame_l assertion; otherwise, the 21554 terminates the transaction with a master abort. Upon completion of a transaction, p_devsel_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
p_frame_l	STS	Primary PCI interface FRAME#. Signal p_frame_l is driven by the initiator of a transaction to indicate the beginning and duration of an access on the primary PCI bus. Signal p_frame_l assertion (falling edge) indicates the beginning of a PCI transaction. While p_frame_l remains asserted, data transfers can continue. The deassertion of p_frame_l indicates the final data phase requested by the initiator. Upon completion of a transaction, p_frame_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
p_gnt_l	I	Primary PCI bus GNT#. When asserted, p_gnt_l indicates to the 21554 that access to the primary bus is granted. The 21554 can start a transaction on the primary bus when the bus is idle and p_gnt_l is asserted. When the 21554 has not requested use of the bus and p_gnt_l is asserted, the 21554 drives p_ad, p_cbe_l, and p_par to valid logic levels.
p_idsel	I	Primary PCI interface IDSEL. Signal p_idsel is used as the chip select line for Type 0 configuration accesses to 21554 configuration space from the primary bus. When p_idsel is asserted during the address phase of a Type 0 configuration transaction, the 21554 responds to the transaction by asserting p_devsel_l.
p_inta_l	OD	Primary PCI bus interrupt. Signal p_inta_l is asserted by the 21554 when: • A primary doorbell register bit is set. • The I_2O outbound queue is not empty. • The subsystem event bit is set. All of these conditions are individually maskable. When the corresponding event bit is cleared or the I_2O outbound queue is emptied, p_inta_l is deasserted. Signal p_inta_l is pulled up through an external resistor.

Table 3-3. Primary PCI Bus Interface Signals (Sheet 2 of 3)

Signal Name	Type	Description
p_irdy_l	STS	Primary PCI interface IRDY#. Signal p_irdy_l is driven by the initiator of a transaction to indicate the initiator's ability to complete the current data phase on the primary PCI bus. During a write transaction, assertion of p_irdy_l indicates that valid write data is being driven on the p_ad bus. During a read transaction, assertion of p_irdy_l indicates that the initiator is able to accept read data for the current data phase. Once asserted during a given data phase, p_irdy_l is not deasserted until the data phase completes. Upon completion of a transaction, p_irdy_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
p_par	TS	Primary PCI interface parity. Signal p_par carries the even parity of the 36 bits of p_ad[31:0] and p_cbe_l[3:0] for both address and data phases. Signal p_par is driven by the same agent that drives the address (for address parity) or the data (for data parity). Signal p_par contains valid parity one clock cycle after the address is valid (indicated by assertion of p_frame_l), or one clock cycle after the data is valid (indicated by assertion of p_irdy_l for write transactions and p_trdy_l for read transactions). Signal p_par is tristated one clock cycle after the p_ad lines are tristated. The device receiving data samples p_par as an input to check for possible parity errors. When the primary PCI bus is idle, the 21554 drives p_par to a valid logic level when p_gnt_l is asserted (one clock cycle after the p_ad bus is parked).
p_perr_l	STS	Primary PCI interface PERR#. Signal p_perr_l is asserted when a data parity error is detected for data received on the primary interface. The timing of p_perr_l corresponds to p_par driven one clock cycle earlier, and p_ad and p_cbe_l driven two clock cycles earlier. Signal p_perr_l is asserted by the target during write transactions, and by the initiator during read transactions. Upon completion of a transaction, p_perr_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
p_req_l	TS	Primary PCI bus REQ#. Signal p_req_l is asserted by the 21554 to indicate to the primary bus arbiter that it wants to start a transaction on the primary bus.
p_serr_l	OD	Primary PCI interface SERR#. Signal p_serr_l can be driven low by any device on the primary bus to indicate a system error condition. The 21554 can conditionally assert p_serr_l for the following reasons: • Primary bus address parity error • Downstream posted write data parity error on secondary bus • Master abort during downstream posted write transaction • Target abort during downstream posted write transaction • Downstream posted write transaction discarded • Downstream delayed write request discarded • Downstream delayed read request discarded • Downstream delayed transaction master timeout • Secondary bus s_serr_l assertion Signal p_serr_l is pulled up through an external resistor.

Table 3-3. Primary PCI Bus Interface Signals (Sheet 3 of 3)

Signal Name	Type	Description
p_stop_l	STS	Primary PCI interface STOP#. Signal p_stop_l is driven by the target of a transaction, indicating that the target is requesting the initiator to stop the transaction on the primary bus. • When p_stop_l is asserted in conjunction with p_trdy_l and p_devsel_l assertion, a disconnect with data transfer is being signaled. • When p_stop_l and p_devsel_l are asserted, but p_trdy_l is deasserted, a target disconnect without data transfer is being signaled. When this occurs on the first data phase, that is, no data is transferred during the transaction, this is referred to as a target retry. • When p_stop_l is asserted and p_devsel_l is deasserted, the target is signaling a target abort. Upon completion of a transaction, p_stop_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
p_trdy_l	STS	Primary PCI interface TRDY#. Signal p_trdy_l is driven by the target of a transaction to indicate the target's ability to complete the current data phase on the primary PCI bus. During a write transaction, assertion of p_trdy_l indicates that the target is able to accept write data for the current data phase. During a read transaction, assertion of p_trdy_l indicates that the target is driving valid read data on the p_ad bus. Once asserted during a given data phase, p_trdy_l is not deasserted until the data phase completes. Upon completion of a transaction, p_trdy_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.

3.2 Primary PCI Bus Interface 64-Bit Extension Signals

Table 3-4 describes the primary PCI bus interface 64-bit extension signals.

Table 3-4. Primary PCI Bus Interface 64-Bit Extension Signals

Signal Name	Type	Description
p_ack64_l	STS	Primary PCI interface acknowledge 64-bit transfer. Signal p_ack64_l is asserted by the target only when p_req64_l is asserted by the initiator, to indicate the target's ability to transfer data using 64 bits. Signal p_ack64_l has the same timing as p_devsel_l. When deasserting, p_ack64_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
p_ad[63:32]	TS	Primary PCI interface address/data upper 32 bits. This multiplexed address and data bus provides an additional 32 bits to the primary interface. During the address phase or phases of a transaction, when the dual-address command is used and p_req64_l is asserted, the initiator drives the upper 32 bits of a 64-bit address; otherwise, these bits are undefined, and the initiator drives a valid logic level onto the pins. During the data phases of a transaction, the initiator drives the upper 32 bits of 64-bit write data, or the target drives the upper 32 bits of 64-bit read data, when p_req64_l and p_ack64_l are both asserted. When not driven, signals p_ad[63:32] are pulled up to a valid logic level through external resistors.
p_cbe_l[7:4]	TS	Primary PCI interface command/byte enables upper 4 bits. These signals are a multiplexed command field and byte enable field. During the address phase or phases of a transaction, when the dual-address command is used and p_req64_l is asserted, the initiator drives the transaction type on p_cbe_l[7:4]; otherwise, these bits are undefined, and the initiator drives a valid logic level onto the pins. For both read and write transactions, the initiator drives byte enables for the p_ad[63:32] data bits on p_cbe_l[7:4] during the data phases, when p_req64_l and p_ack64_l are both asserted. When not driven, signals p_cbe_l[7:4] are pulled up to a valid logic level through external resistors.
p_par64	TS	Primary PCI interface upper 32 bits parity. Signal p_par64 carries the even parity of the 36 bits of p_ad[63:32] and p_cbe_l[7:4] for both address and data phases. Signal p_par64 is driven by the initiator and is valid one clock cycle after the first address phase when a dual-address command is used and p_req64_l is asserted. Signal p_par64 is also valid one clock cycle after the second address phase of a dual-address transaction when p_req64_l is asserted. Signal p_par64 is valid one clock cycle after valid data is driven (indicated by assertion of p_irdy_l for write data and p_trdy_l for read data), when both p_req64_l and p_ack64_l are asserted for that data phase. Signal p_par64 is tristated by the device driving read or write data one clock cycle after the p_ad lines are tristated. Devices receiving data sample p_par64 as an input to check for possible parity errors during 64-bit transactions. When not driven, p_par64 is pulled up to a valid logic level through external resistors.
p_req64_l	STS	Primary PCI interface request 64-bit transfer. Signal p_req64_l is asserted by the initiator to indicate that the initiator is requesting 64-bit data transfer. Signal p_req64_l has the same timing as p_frame_l. When deasserting, p_req64_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor. The 21554 samples p_req64_l during primary bus reset to enable the 64-bit extension signals. If p_req64_l is sampled high during reset, the primary 64-bit extension is disabled and assumed not connected. The 21554 then drives p_ad[63:32], p_cbe_l[7:4], and p_par64 to valid logic levels.

3.3 Secondary PCI Bus Interface Signals

Table 3-5 describes the secondary PCI bus interface signals.

Table 3-5. Secondary PCI Bus Interface Signals (Sheet 1 of 2)

Signal Name	Type	Description
s_ad[31:0]	TS	Secondary PCI interface address/data. These signals are a 32-bit multiplexed address and data bus. During the address phase or phases of a transaction, the initiator drives a physical address on s_ad[31:0]. During the data phases of a transaction, the initiator drives write data, or the target drives read data, on s_ad[31:0]. When the secondary PCI bus is idle, the 21554 drives s_ad to a valid logic level when its secondary bus grant is asserted.
s_cbe_l[3:0]	TS	Secondary PCI interface command/byte enables. These signals are a multiplexed command field and byte enable field. During the address phase or phases of a transaction, the initiator drives the transaction type on s_cbe_l[3:0]. When there are two address phases, the first address phase carries the dual-address command and the second address phase carries the transaction type. For both read and write transactions, the initiator drives byte enables on s_cbe_l[3:0] during the data phases. When the secondary PCI bus is idle, the 21554 drives s_cbe_l to a valid logic level when its secondary bus grant is asserted.
s_devsel_l	STS	Secondary PCI interface DEVSEL#. Signal s_devsel_l is asserted by the target, indicating that the device is responding to the transaction. As a target, the 21554 decodes the address of a transaction initiated on the secondary bus to determine whether to assert s_devsel_l. As an initiator of a transaction on the secondary bus, the 21554 looks for the assertion of s_devsel_l within five clock cycles of s_frame_l assertion; otherwise, the 21554 terminates the transaction with a master abort. Upon completion of a transaction, s_devsel_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
s_frame_l	STS	Secondary PCI interface FRAME#. Signal s_frame_l is driven by the initiator of a transaction to indicate the beginning and duration of an access on the secondary PCI bus. Signal s_frame_l assertion (falling edge) indicates the beginning of a PCI transaction. While s_frame_l remains asserted, data transfers can continue. The deassertion of s_frame_l indicates the final data phase requested by the initiator. Upon completion of a transaction, s_frame_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
s_idsel	I	Secondary PCI interface IDSEL. Signal s_idsel is used as the chip select line for Type 0 configuration accesses to 21554 configuration space from the secondary bus. When s_idsel is asserted during the address phase of a Type 0 configuration transaction, the 21554 responds to the transaction by asserting s_devsel_l.
s_inta_l	OD	Secondary PCI bus interrupt. Signal s_inta_l is asserted by the 21554 when: • A secondary doorbell register bit is set. • The I_2O inbound queue is not empty. • A page boundary is reached when performing lookup table address translation. • The 21554 transitions from a D1 or D2 power state to a D0 power state. All of these conditions are individually maskable. Signal s_inta_l is deasserted when the corresponding event bit is cleared, or when the I_2O inbound queue is empty. Signal s_inta_l is pulled up through an external resistor.
s_irdy_l	STS	Secondary PCI interface IRDY#. Signal s_irdy_l is driven by the initiator of a transaction to indicate the initiator's ability to complete the current data phase on the secondary PCI bus. During a write transaction, assertion of s_irdy_l indicates that valid write data is being driven on the s_ad bus. During a read transaction, assertion of s_irdy_l indicates that the initiator is able to accept read data for the current data phase. Once asserted during a given data phase, s_irdy_l is not deasserted until the data phase completes. Upon completion of a transaction, s_irdy_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.

Table 3-5. Secondary PCI Bus Interface Signals (Sheet 2 of 2)

Signal Name	Type	Description
s_par	TS	Secondary PCI interface parity. Signal s_par carries the even parity of the 36 bits of s_ad[31:0] and s_cbe_l[3:0] for both address and data phases. Signal s_par is driven by the same agent that drives the address (for address parity) or the data (for data parity). Signal s_par contains valid parity one clock cycle after the address is valid (indicated by assertion of s_frame_l), or one clock cycle after the data is valid (indicated by assertion of s_irdy_l for write transactions and s_trdy_l for read transactions). Signal s_par is tristated one clock cycle after the s_ad lines are tristated. The device receiving data samples s_par as an input to check for possible parity errors. When the secondary PCI bus is idle, the 21554 drives s_par to a valid logic level when its secondary bus grant is asserted (one clock cycle after the s_ad bus is parked).
s_perr_l	STS	Secondary PCI interface PERR#. Signal s_perr_l is asserted when a data parity error is detected for data received on the secondary interface. The timing of s_perr_l corresponds to s_par driven one clock cycle earlier, and s_ad driven two clock cycles earlier. Signal s_perr_l is asserted by the target during write transactions, and by the initiator during read transactions. Upon completion of a transaction, s_perr_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
s_serr_l	OD	Secondary PCI interface SERR#. Signal s_serr_l can be driven low by any device on the secondary bus to indicate a system error condition. The 21554 also samples s_serr_l as an input and conditionally forwards it to the primary bus on p_serr_l. The 21554 can conditionally assert s_serr_l for the following reasons: • Secondary bus address parity error • Upstream posted write data parity error on primary bus • Master abort during upstream posted write transaction • Target abort during upstream posted write transaction • Upstream posted write transaction discarded • Upstream delayed write request discarded • Upstream delayed read request discarded • Upstream delayed transaction master timeout Signal s_serr_l is pulled up through an external resistor.
s_stop_l	STS	Secondary PCI interface STOP#. Signal s_stop_l is driven by the target of a transaction, indicating that the target is requesting the initiator to stop the transaction on the secondary bus. • When s_stop_l is asserted in conjunction with s_trdy_l and s_devsel_l assertion, a disconnect with data transfer is being signaled. • When s_stop_l and s_devsel_l are asserted, but s_trdy_l is deasserted, a target disconnect without data transfer is being signaled. When this occurs on the first data phase, that is, no data is transferred during the transaction, this is referred to as a target retry. • When s_stop_l is asserted and s_devsel_l is deasserted, the target is signaling a target abort. Upon completion of a transaction, s_stop_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
s_trdy_l	STS	Secondary PCI interface TRDY#. Signal s_trdy_l is driven by the target of a transaction to indicate the target's ability to complete the current data phase on the secondary PCI bus. During a write transaction, assertion of s_trdy_l indicates that the target is able to accept write data for the current data phase. During a read transaction, assertion of s_trdy_l indicates that the target is driving valid read data on the s_ad bus. Once asserted during a given data phase, s_trdy_l is not deasserted until the data phase completes. Upon completion of a transaction, s_trdy_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.

3.4 Secondary PCI Bus Interface 64-Bit Extension Signals

Table 3-6 describes the secondary PCI bus interface 64-bit extension signals.

Table 3-6. Secondary PCI Bus Interface 64-Bit Extension Signals

Signal Name	Type	Description
s_ack64_l	STS	Secondary PCI interface acknowledge 64-bit transfer. Signal s_ack64_l is asserted by the target only when s_req64_l is asserted by the initiator, to indicate the target's ability to transfer data using 64 bits. Signal s_ack64_l has the same timing as s_devsel_l. When deasserting, s_ack64_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor.
s_ad[63:32]	TS	Secondary PCI interface address/data upper 32 bits. This multiplexed address and data bus provides an additional 32 bits to the secondary interface. During the address phase or phases of a transaction, when the dual-address command is used and s_req64_l is asserted, the initiator drives the upper 32 bits of a 64-bit address; otherwise, these bits are undefined, and the initiator drives a valid logic level onto the pins. During the data phases of a transaction, the initiator drives the upper 32 bits of 64-bit write data, or the target drives the upper 32 bits of 64-bit read data, when s_req64_l and s_ack64_l are both asserted. When not driven, signals s_ad[63:32] are pulled up to a valid logic level through external resistors.
s_cbe_l[7:4]	TS	Secondary PCI interface command/byte enables upper 4 bits. These signals are a multiplexed command field and byte enable field. During the address phase or phases of a transaction, when the dual-address command is used and s_req64_l is asserted, the initiator drives the transaction type on s_cbe_l[7:4]; otherwise, these bits are undefined, and the initiator drives a valid logic level onto the pins. For both read and write transactions, the initiator drives byte enables for the s_ad[63:32] data bits on s_cbe_l[7:4] during the data phases, when s_req64_l and s_ack64_l are both asserted. When not driven, signals s_cbe_l[7:4] are pulled up to a valid logic level through external resistors.
s_par64	TS	Secondary PCI interface upper 32 bits parity. Signal s_par64 carries the even parity of the 36 bits of s_ad[63:32] and s_cbe_l[7:4] for both address and data phases. Signal s_par64 is driven by the initiator and is valid one clock cycle after the first address phase when a dual-address command is used and s_req64_l is asserted. Signal s_par64 is also valid one clock cycle after the second address phase of a dual-address transaction when s_req64_l is asserted. Signal s_par64 is valid one clock cycle after valid data is driven (indicated by assertion of s_irdy_l for write data and s_trdy_l for read data), when both s_req64_l and s_ack64_l are asserted for that data phase. Signal s_par64 is tristated by the device driving read or write data one clock cycle after the s_ad lines are tristated. Devices receiving data sample s_par64 as an input to check for possible parity errors during 64-bit transactions. When not driven, s_par64 is pulled up to a valid logic level through external resistors.
s_req64_l	STS	Secondary PCI interface request 64-bit transfer. Signal s_req64_l is asserted by the initiator to indicate that the initiator is requesting 64-bit data transfer. Signal s_req64_l has the same timing as s_frame_l. If the 21554 is the secondary bus central function, it will assert s_req64_l low during secondary bus reset to indicate that a 64-bit bus is supported. When deasserting, s_req64_l is driven to a deasserted state for one clock cycle and is then sustained by an external pull-up resistor. The 21554 samples s_req64_l during secondary bus reset to enable the 64-bit extension signals. If s_req64_l is sampled high during reset, the secondary 64-bit extension is disabled and assumed not connected. The 21554 then drives s_ad[63:32], s_cbe_l[7:4], and s_par64 to valid logic levels.

3.5 Secondary PCI Bus Arbitration Signals

Table 3-7 describes the secondary PCI bus arbitration signals.

Table 3-7. Secondary PCI Bus Arbitration Signals

Signal Name	Type	Description
s_gnt_l[8:0]	TS	Secondary PCI interface GNT#s. The 21554 secondary bus arbiter can assert one of nine secondary bus grant outputs, s_gnt_l[8:0], to indicate that an initiator can start a transaction on the secondary bus if the bus is idle. The 21554's secondary bus grant is an internal signal. A programmable two-level rotating priority algorithm is used. If the internal arbiter is disabled, s_gnt_l[0] is reconfigured to be an external secondary bus request output for the 21554. The 21554 asserts this signal whenever it wants to start a transaction on the secondary bus.
s_req_l[8:0]	I	Secondary PCI interface REQ#s. The 21554 accepts nine request inputs, s_req_l[8:0], into its secondary bus arbiter. The 21554 request input to the arbiter is an internal signal. Each request input can be programmed to be in either a high-priority rotating group or a low-priority rotating group. An asserted level on an s_req_l pin indicates that the corresponding master wants to initiate a transaction on the secondary PCI bus. If the internal arbiter is disabled, s_req_l[0] is reconfigured to be an external secondary grant input for the 21554. In this case, an asserted level on s_req_l[0] indicates that the 21554 can start a transaction on the secondary PCI bus if the bus is idle.

3.6 Clock Signals

Table 3-8 describes the clock signals.

Table 3-8. Clock Signals

Signal Name	Type	Description
p_clk	I	Primary interface PCI CLK. This signal provides timing for all transactions on the primary PCI bus. All primary PCI inputs are sampled on the rising edge of p_clk, and all primary PCI outputs are driven from the rising edge of p_clk. Frequencies supported by the 21554 range from 0 MHz to 33 MHz.
s_clk	I	Secondary interface PCI CLK. This signal provides timing for all transactions on the secondary PCI bus. All secondary PCI inputs are sampled on the rising edge of s_clk, and all secondary PCI outputs are driven from the rising edge of s_clk. Frequencies supported by the 21554 range from 0 MHz to 33 MHz.
s_clk_o	O	Secondary interface PCI CLK output. This signal is generated from the primary interface clock input, p_clk. This clock operates at the same frequency of p_clk and may be externally buffered to create secondary bus device clock signals. When buffered clocks are used, one of the clock outputs must be fed back to the secondary clock input, s_clk. This clock output can be disabled by writing the secondary clock disable bit in configuration space, or by pulling pr_ad[5] low during reset.

3.7 Power Management, Hot Swap, and Reset Signals

Table 3-9 describes the power management, hot swap, and reset signals.

Table 3-9. Power Management, Hot Swap, and Reset Signals

Signal Name	Type	Description
l_stat	TS	Compact PCI hot swap local status pin. As an input to the 21554, this signal indicates the sense of the ejector switch and therefore the state of the LED in a Compact PCI card supporting distributed hot-swap. As an output from the 21554, it controls the LED. If compact PCI hot swap is not supported by the add-in card, this signal should be tied low through a resistor.
p_enum_l	OD	Primary bus compact PCI hot swap event. Conditionally asserted by the 21554, this signal indicates either that the card has been inserted and is ready for configuration, or that the card is about to be removed. This signal is deasserted when the corresponding insertion or removal event bit is cleared. This signal should be pulled up by an external resistor.
p_pme_l	OD	Primary bus power management event. Provides power management signaling capability on behalf of the subsystem. The 21554 asserts p_pme_l when all of the following are true: • Signal s_pme_l is asserted low. • Signal p_pme_l is supported in the current power state. • PME_EN bit is set. Once asserted, p_pme_l is deasserted when the PME status bit or the PME_EN bit is cleared. If the PME# isolation circuitry is needed, it must be implemented externally.
p_rst_l	I	Primary PCI bus RST#. Signal p_rst_l forces the 21554 to a known state. All register state is cleared, and all PCI bus outputs are tristated, with the possible exception of s_ad, s_cbe_l, s_par, and s_req64_l. Signal p_rst_l is asynchronous to p_clk.
s_pme_l	I	Secondary bus power management event. The subsystem asserts this signal to the 21554 to indicate that it is signaling a power management event. The 21554 conditionally asserts p_pme_l when s_pme_l is asserted low. If the subsystem does not generate power management events, this signal can also be used for a subsystem status signal. A deasserting (rising) edge on this signal may conditionally cause the 21554 to assert p_inta_l. If this signal is not used, it should be tied high through a resistor.
s_rst_l	O	Secondary PCI bus RST#. Signal s_rst_l is driven by the 21554 and acts as the PCI reset for the secondary bus. The 21554 asserts s_rst_l when any of the following conditions is met: • Signal p_rst_l is asserted. • The secondary reset bit in the reset control register in configuration space is set. • The chip reset bit in the reset control register in configuration space is set. • Power management transition from $D3_{hot}$ to D0 occurs. When the 21554 asserts s_rst_l, it tristates all secondary control signals and, if designated as the secondary bus central resource, asserts s_req64_l and drives zeros on s_ad, s_cbe_l, and s_par. Signal s_rst_l remains asserted until p_rst_l is deasserted, and the secondary reset bit is clear. Deassertion of s_rst_l occurs automatically based on internal timers when s_rst_l assertion is caused by setting the chip reset bit or a power management transition. Assertion of s_rst_l by itself does not clear register state, and configuration registers are still accessible from the primary PCI interface.

3.8 ROM Interface Signals

Table 3-10 describes the ROM interface signals.

Table 3-10. ROM Interface Signals (Sheet 1 of 2)

Signal Name	Type	Description
pr_ad[7:0]	TS	These signals interface to both the serial and parallel external ROM circuitry and have multiple functions. The signals pr_ad[7:0] serve as multiplexed address/data for the parallel ROM and are latched externally in the following sequence: • Address [23:16] • Address [15:8] • Address [7:0] • Data [7:0] The signals pr_ad[2:0] also serve as serial ROM signals, with no external logic required: • pr_ad[2] : sr_do, the serial ROM data output • pr_ad[1] : sr_di, the serial ROM data input • pr_ad[0] : sr_ck, the serial ROM clock output During primary bus reset, external pull-up or pull-down resistors can be used on signals pr_ad[7:2] to specify their state during reset. The values of these signals during primary bus reset specify the following configuration options: pr_ad[7] During primary bus reset, pr_ad[7] specifies the arbiter enable configuration. If low, the secondary bus arbiter is disabled, s_gnt_l[0] is used for 21554 secondary bus request, and s_req_l[0] is used for 21554 secondary bus grant. If high, the internal arbiter is enabled for use. pr_ad[6] During primary bus reset, pr_ad[6] specifies the central function enable. If low, the 21554 asserts s_req64_l and drives s_ad, s_cbe_l, and s_par low during secondary reset. If high, the 21554 tristates s_req64_l, s_ad, s_cbe_l, and s_par during secondary reset. pr_ad[5] During primary bus reset, pr_ad[5] specifies the s_clk_o enable. If low, s_clk_o is disabled and driven low. If high, s_clk_o is enabled and is a buffered version of p_clk. pr_ad[4] During primary bus reset, pr_ad[4] specifies synchronous enable. If high, the 21554 assumes asynchronous primary and secondary interfaces. If low, the 21554 assumes synchronous primary and secondary interfaces. pr_ad[3] During primary bus reset, pr_ad[3] specifies the primary lockout bit reset value. If high, the primary lockout bit is set high upon completion of chip reset, which causes the 21554 to return target retry to primary bus configuration transactions until the bit is cleared. If low, the primary lockout bit is low upon completion of reset, which allows immediate primary bus access to configuration registers. pr_ad[2] This signal should be biased high through a pull-up resistor. If the serial ROM is not connected, the 21554 will not detect the pre-load enable sequence 10b. In this case, the serial ROM preload is terminated after the first bit is read and the 21554 registers remain at their reset values. This is not actually sampled at reset, but during the first serial ROM read.

Table 3-10. ROM Interface Signals (Sheet 2 of 2)

Signal Name	Type	Description
pr_ale_l	O	Parallel ROM address latch enable/chip select decoder enable. The signal pr_ale_l is used to enable the parallel ROM address latches. The 21554 asserts pr_ale_l low when it drives the first eight bits of the 24-bit address on pr_ad[7:0], and keeps it asserted until the last eight bits of the address are driven. The upper bits of the address are shifted through octal D-registers while pr_ale_l is low. When in multiple device mode, pr_ale_l is also used for a chip select enable. When pr_ale_l is high, the upper latched address lines are decoded with external circuitry to assert device chip enables.
pr_clk	O	Parallel ROM address latch clock output. The signal pr_clk is used to clock the address registers needed to de-multiplex the address, and is a buffered version of p_clk divided by two.
pr_cs_l	O/I	Parallel ROM chip select or device ready. For a single device attachment, pr_cs_l is used for the parallel ROM chip select. The 21554 asserts pr_cs_l low after the address is shifted out and de-multiplexed through external octal registers. The 21554 deasserts pr_cs_l according to the access time specified in the ROM control CSR. When in multiple device mode, pr_cs_l is reconfigured as a device ready (pr_rdy) input. If pr_cs_l is driven low while the read or write strobe is asserted, the assertion time of the read or write strobe is extended by the amount of time the device ready signal is held low.
pr_rd_l	O	Parallel ROM read strobe. This signal controls the output enable signal of the parallel ROM. The 21554 asserts pr_rd_l to enable the ROM to drive read data on pr_ad[7:0]. The 21554 samples this read data on the deasserting (rising) edge of pr_rd_l. The timing of pr_rd_l with respect to the chip select is dictated by the read strobe mask.
pr_wr_l	O	Parallel ROM write strobe. This signal controls the write enable signal of the parallel ROM. The 21554 asserts pr_wr_l when it drives write data to the ROM on pr_ad[7:0]. Write data is held stable until the deasserting (rising) edge of pr_wr_l. The timing of pr_wr_l with respect to the chip select is dictated by the write strobe mask.
sr_cs	O	Serial ROM chip select. The 21554 drives this signal high to enable the serial ROM for a read or write. The serial ROM operation uses pins pr_ad[2:0] for data in, data out, and clock.

3.9 Miscellaneous Signals

Table 3-11 describes the miscellaneous signals.

Table 3-11. Miscellaneous Signals

Signal Name	Type	Description
p_vio	I	Primary interface I/O voltage. This signal must be tied to either 3.3 V or 5.0 V, corresponding to the signaling environment of the primary PCI bus as described in the *PCI Local Bus Specification*, *Revision 2.1*. When any device on the primary PCI bus uses 5-V signaling levels, tie p_vio to 5.0 V. Signal p_vio is tied to 3.3 V only when all the devices on the primary bus use 3.3-V signaling levels.
s_vio	I	Secondary interface I/O voltage. This signal must be tied to either 3.3 V or 5.0 V, corresponding to the signaling environment of the secondary PCI bus as described in the *PCI Local Bus Specification, Revision 2.1*. When any device on the secondary PCI bus uses 5-V signaling levels, tie s_vio to 5.0 V. Signal s_vio is tied to 3.3 V only when all the devices on the secondary bus use 3.3-V signaling levels.

3.10 Diagnostic Signals

Table 3-12 describes JTAG and other diagnostic signals.

Table 3-12. JTAG Signals

Signal Name	Type	Description
scan_ena	I	Scan enable input. This signal is used for chip test only and should be tied low through an external resistor.
tck	I	JTAG boundary-scan clock. Signal tck is the clock controlling the JTAG logic.
tdi	I	JTAG serial data in. Signal tdi is the serial input through which JTAG instructions and test data enter the JTAG interface. The new data on tdi is sampled on the rising edge of tck. An unterminated tdi produces the same result as if tdi were driven high.
tdo	O	JTAG serial data out. Signal tdo is the serial output through which test instructions and data from the test logic leave the 21554.
tms	I	JTAG test mode select. Signal tms causes state transitions in the test access port (TAP) controller. An undriven tms has the same result as if it were driven high.
trst_l	I	JTAG TAP reset. When asserted low, the TAP controller is asynchronously forced to enter a reset state, which in turn asynchronously initializes other test logic. An unterminated trst_l produces the same result as if trst_l were driven high.

PCI Bus Operation 4

This chapter presents detailed information about PCI transactions, transaction forwarding across the 21554, and transaction termination.

4.1 PCI Bus Operation

The 21554 responds to transactions using the following commands as a target on both interfaces:

- All memory commands
- I/O read and write commands
- Dual-address commands
- Type 0 and Type 1 configuration commands

The 21554 does not respond to transactions using any other PCI commands.

The 21554 can initiate transactions using the following commands on either interface:

- All memory commands
- I/O read and write commands
- Dual-address commands
- Type 0 configuration commands

The 21554 does not initiate transactions using any other PCI commands.

The 21554 responds to transactions by asserting DEVSEL# with medium timing. The 21554 may also be enabled to subtractively decode I/O transactions in one direction only.

The 21554 supports linear increment address mode only, and disconnects memory transactions whose low two address bits are not 00b after a single Dword.

4.1.1 Posted Write Transactions

The 21554 posts all memory write and memory write and invalidate (MWI) transactions that are to be forwarded from one interface to the other. The 21554 accepts write data into its buffers without wait states until the initiator ends the transaction, until an aligned address boundary is reached, or until the posted write queue fills. Aligned address disconnect boundaries for memory write and MWI transactions are listed in Section 4.1.1.1 and Section 4.1.1.2, respectively.

The 21554 does not initiate a memory write transaction on the target bus until at least a cache line amount of data is posted. If the transaction consists of less than a cache line, the 21554 waits until the entire burst is posted. If the cache line size corresponding to the target bus is not set to a valid value, then the 21554 uses a value of 8 Dwords for this purpose. Possible valid values are 4, 8, 16 and 32 Dwords.

The 21554 continues the transaction to the target as long as write data is available or the transaction has terminated on the initiator bus. Otherwise, the 21554 ends the transaction when a queue-empty condition is detected or when all write data has been delivered for this transaction. The 21554 does not insert master wait states when initiating posted writes.

If the 21554 receives 2^{24} consecutive target retries from the target when attempting to deliver posted write data, then the 21554 discards the posted write transaction and conditionally asserts SERR# on the initiator bus (see Chapter 14). The 21554 also conditionally asserts SERR# on the initiator bus if a target abort or master abort is detected on the target bus in response to the posted write.

4.1.1.1 Memory Write Transactions

As a target, the 21554 disconnects memory write transactions at the following address boundaries:

- An aligned 4KB address boundary
- An aligned page address boundary, for upstream transactions falling in the Upstream Memory 2 address range
- An aligned cache line boundary, when memory write Disconnect bit is set in configuration space

If the posted write queue fills before the master terminates the transaction, then the 21554 returns a target disconnect when the last queue entry is filled.

As an initiator, when the 21554 has posted write data to deliver and the conditions listed in Section 4.1.1.2 for initiating an MWI transaction are not met, the 21554 uses the memory write command to deliver posted memory write data.

4.1.1.2 Memory Write and Invalidate Transactions

As a target, the 21554 disconnects MWI transactions at the following address boundaries:

- An aligned 4 KB address boundary
- An aligned page address boundary, for upstream transactions falling in the Upstream Memory 2 address range
- An aligned cache line boundary, for MWI transactions when less than a cache line of available space remains in the posted write queue

When a master initiates an MWI transaction, it guarantees that it will supply one full cache line of data, or some multiple thereof. The 21554 initiates an MWI transaction on the target bus, regardless of whether the bus command was a memory write or an MWI on the initiator bus, whenever all of the following conditions are met:

- The MWI Enable bit is set in the Command register corresponding to the target interface.
- The target bus Cache Line Size is set to a valid value (4, 8, 16, or 32 Dwords).
- At least one aligned cache line of data has been posted.
- All byte enables for the posted cache line are turned on.

If any of these conditions is not met, the 21554 uses the memory write command.

The 21554 continues the MWI transaction as long as a full cache line (corresponding to the cache line size of the target bus) is posted in the posted write queue. If this condition does not exist, then the 21554 master terminates the transaction at the cache line boundary.

If the 21554 terminates an MWI transaction before all write data is delivered, then the 21554 initiates another write transaction to finish delivery of the write data. If a fraction of a cache line remains, the 21554 initiates the transaction with the memory write command. If at least a complete cache line was subsequently posted, then the 21554 once again initiates the transaction with an MWI command.

4.1.1.3 Posted Write Transactions Using the 64-bit Extension

The 21554 uses the 64-bit extension signals for accepting and delivering posted write data.

As a target, the 21554 asserts ACK64# in response to the initiator's assertion of REQ64# for memory writes and MWI commands if the address is quadword aligned (address bit AD[2] is zero). The 21554 then accepts 64 bits of data per data phase without inserting target wait states.

As an initiator, the 21554 asserts REQ64# when delivering posted write data as long as the burst consists of a minimum of 4 Dwords, and the original address is quadword aligned. If the target asserts ACK64#, write data is delivered 64 bits per data phase without inserting master wait states. If the burst ends on an odd Dword address boundary, the 21554 forces the high four byte enables of the last data phase in the burst to be deasserted.

4.1.1.4 Write Performance Tuning Options

The 21554 implements several features and options that affect write performance when forwarding posted write transactions.

Memory Write and Invalidate

When the MWI Enable bit in configuration space is set for that corresponding interface, the 21554 is enabled to initiate MWI transactions as described in Section 4.1.1.2.

Fast Back-to-Back

The 21554 may be enabled to initiate fast back-to-back transactions. The 21554 must have the bus grant the clock cycle before it asserts FRAME# for the second transaction, and the Fast Back-to-Back Enable bit must be set for the interface on which the 21554 is initiating the transaction. If both of these conditions exist, the 21554 may initiate the second transaction with fast back-to-back timing following a write transaction that is not terminated with STOP#.

Write Flow-Through

If the 21554 is able to obtain access to the target bus and start transferring write data to the target before the transaction has been terminated on the initiator bus, then the 21554 automatically enters flow-through mode. When in flow-through mode, the 21554 can sustain long write bursts as long as a queue-empty condition is not detected, or until an aligned disconnect boundary is reached. If the queue-empty condition is detected, the 21554 master terminates the transaction on the target bus. If an aligned disconnect boundary is reached, the 21554 returns a target disconnect on the initiator bus. Flow-through mode behavior is used for both memory write and MWI commands.

Memory Write Disconnect Mode

The 21554 implements a Memory Write Disconnect Mode bit in device specific configuration space. When enabled, the 21554 disconnects memory writes on aligned cache line boundaries, using the cache line size corresponding to the target bus.

Posted Write Queue Tuning

The 21554 implements a posted write queue management control bit for each posted write queue in the Chip Control 1 configuration register. This bit specifies at what threshold the 21554 returns a target retry instead of accepting write data. Setting this bit can minimize fragmentation of posted write transactions and can prevent bursts from being broken into sub-cache line bursts. The tuning options are as follows:

- Target retry is returned when less then a cache line is free.
- Target retry is returned when less then half a cache line is free, for CLS = 8, 16, or 32 Dwords. A full cache line threshold is used for CLS = 4.

4.1.2 Delayed Write Transactions

The 21554 uses delayed transactions when forwarding I/O writes from one PCI interface to the other. Delayed transactions are also used for CSR or configuration register writes that cause the 21554 to initiate a transaction on the opposite interface, such as:

- CSR or configuration register write access that causes the 21554 to initiate a configuration write transaction
- CSR write access that causes the 21554 to initiate an I/O write transaction

When an I/O write intended for the opposite PCI bus is first initiated, the 21554 returns a target retry. If the delayed transaction queue is not full, the 21554 queues the transaction information, including address, bus command, write data, and byte enables.

If the transaction queued is a result of an I/O or Configuration Data register write, the 21554 queues the appropriate data based on the type of access desired, the address and data contained in the corresponding registers, and the byte enables used for the register access. This phase of the delayed transaction is called a delayed write request (DWR).

The 21554 requests the target bus and initiates the delayed write transaction as soon as the 21554 ordering rules allow (see Section 4.1.5). The 21554 always performs a single 32-bit data phase when initiating a delayed write transaction. The 21554 completes the transaction on the target bus and adds the completion status to the queue. Completion status contains the type of termination (TRDY#, target abort, master abort) and whether PERR# assertion was detected. This phase of the delayed transaction is called the delayed write completion (DWC).

If the 21554 receives 2^{24} consecutive target retries from the target, then the 21554 discards the delayed write request and conditionally asserts SERR# on the initiator bus (see Chapter 14). If the transaction is discarded before completion, the 21554 returns a target abort to the initiator.

When the initiator repeats the transaction using the same address, bus command, write data, and byte enables, then the 21554 returns the appropriate target termination when ordering rules allow. Otherwise, the 21554 continues to return target retry. The target terminations (Table 4-1) returned are as follows:

Table 4-1. Delayed Write Transaction Target Termination Returns

Target Bus Response	Initiator Bus Response
TRDY#	TRDY# (and STOP# if multiple data phases requested)
Target abort	Target abort
Master abort	TRDY# (if Master Abort Mode bit = 0) Target abort (if Master Abort Mode bit = 1)

When the 21554 has a delayed completion to return to an initiator, and the initiator does not repeat the transaction before the Master Time-Out Counter for that interface expires, then the 21554 discards the delayed completion transaction. If enabled to do so, the 21554 asserts SERR# on the initiator bus. The Master Time-Out Counter expiration value is either 2^{10} or 2^{15} PCI clock cycles, programmable in the Chip Control 0 configuration register. The Master Time-Out Counter is disabled when the Master Time-Out Disable bit in the Chip Control 0 configuration register is zero.

4.1.3 Delayed Read Transactions

The 21554 uses delayed transactions when forwarding any type of read (I/O, memory, memory read line, and memory read multiple) from one PCI interface to the other. Delayed transactions are also used for parallel ROM reads and CSR or configuration register reads that cause the 21554 to initiate a PCI read transaction, such as:

- CSR or configuration register read access that causes the 21554 to initiate a configuration read transaction
- CSR read access that causes the 21554 to initiate an I/O read transaction

The delayed read transaction protocol is similar to that of delayed write transactions, with the exception that 64-bit transfers may be used for delayed-memory read transactions. When an I/O or memory read intended for the other PCI bus is first initiated, the 21554 returns a target retry. The 21554 queues the transaction information, including address, bus command, byte enables for nonprefetchable reads, if the delayed transaction queue is not full.

If the transaction queued is a result of a CSR or configuration register read, the 21554 queues the appropriate data based on the type of access desired, the address contained in the register, and the byte enables used for the data register access. This phase of the delayed transaction is called a delayed read request (DRR).

The 21554 requests the target bus and initiates the delayed read transaction as soon as the 21554 ordering rules allow. If the transaction is a nonprefetchable read as described in Section 4.1.3.1, the 21554 requests only a single Dword of data. If the transaction is a memory read, the 21554 follows the prefetch rules outlined in Section 4.1.3.2. The 21554 completes the transaction on the target bus and adds the read data and parity to the read data queue and the completion status to the delayed transaction queue. This phase of the delayed transaction is called the delayed read completion (DRC). If the 21554 receives 2^{24} consecutive target retries from the target, then the 21554 discards the delayed read transaction and conditionally asserts SERR# on the initiator bus. If the transaction is discarded before completion, the 21554 returns a target abort to the initiator.

When the initiator repeats the transaction using the same address, bus command, and byte enables, then the 21554 returns the read data, parity, and appropriate target termination when ordering rules allow. For all memory read type transactions, the 21554 aliases the memory read, memory read line, and memory read multiple commands when comparing a transaction in the delayed transaction queue to one initiated on the PCI bus. Therefore, regardless of the exact command used, if the address matches, and both commands are any type of prefetchable memory read, then the 21554 considers it a match. If there is no match or the ordering rules prevent returning the completion at that point, the 21554 returns target retry. The target terminations (Table 4-2) returned are as follows:

Table 4-2. Delayed Read Transaction Target Termination Returns

Target Bus Response	Initiator Bus Response
TRDY#	TRDY# (and STOP# if returning last data and FRAME# is asserted)
Target abort	Target abort
Master abort	TRDY# and FFFFFFFFh (if Master Abort Mode bit = 0) Target abort (if Master Abort Mode bit = 1)

When the 21554 has a delayed completion to return to an initiator, and the initiator does not repeat the transaction before the Master Time-out Counter for that interface expires, then the 21554 discards the delayed completion transaction. The Master Time-out Counter expiration value is either 2^{10} or 2^{15} PCI clock cycles, programmable in the Chip Control 0 configuration register.

4.1.3.1 Nonprefetchable Reads

The following transactions are considered by the 21554 to be nonprefetchable:

- I/O transactions
- Configuration transactions
- Transactions using the memory read command that address a range configured as nonprefetchable
- Primary bus memory reads to the Expansion ROM BAR

When initiating a nonprefetchable read, the 21554 requests only a single Dword of read data from the target. The 21554 uses the same byte enables driven by the initiator of the transaction.

When the 21554 returns the read data to the initiator, it target-disconnects at completion of the first data phase if the initiator is requesting multiple Dwords.

4.1.3.2 Prefetchable Reads

The following transactions are considered by the 21554 to be prefetchable read transactions:

- Transactions using the memory read line command
- Transactions using the memory read multiple command
- Transactions using the memory read command that address a range configured as prefetchable

During a prefetchable read, the 21554 speculatively reads data from the target before the initiator explicitly requests it. The amount of data read depends on the read command, the cache line size corresponding to the initiator bus, and whether the 21554 is in flow-through mode, as described in Table 4-3. The 21554 drives the byte enables to 0h for all data phases, regardless of the byte enables driven by the initiator of the transaction.

When the 21554 returns prefetchable read data to the initiator, it continues to return read data until the master deasserts FRAME# and IRDY# ending the transaction, or until the 21554 runs out of read data and the target disconnect is returned. If the master terminates the transaction, the 21554 discards the unconsumed read data.

4.1.3.3 Prefetchable Read Transactions Using the 64-bit Extension

The 21554 uses the 64-bit extension signals when implemented, to initiate and complete prefetchable read transactions.

As a target, the 21554 asserts ACK64# in response to the initiator's assertion of REQ64# for prefetchable memory read transactions where the 21554 has more than 1 Dword of data to return. The 21554 returns 64 bits of data per data phase without inserting target wait states, with the exception of a temporary queue-empty condition during flow-through. If the 21554 has an odd number of Dwords to return to the initiator, it disconnects before delivering the last Dword. The last Dword is discarded.

As an initiator, the 21554 asserts REQ64# for all prefetchable memory reads that have a starting address on an aligned quadword boundary (that is, address bit AD[2] = 0). The 21554 then accepts 64 bits of read data per data phase without inserting master wait states.

4.1.3.4 Read Performance Features and Tuning Options

The 21554 implements several features and options that affect read performance when forwarding prefetchable read transactions.

Read Flow-Through

When the bandwidth of the initiator PCI interface is less than or equal to the bandwidth of the target PCI interface, the 21554 may use flow-through, or streaming, operation when returning read data. If the initiator of a delayed prefetchable read transaction repeats the transaction, and the 21554 starts delivering read data on the initiator bus while it is still accepting data for that transaction on the target bus, the 21554 enters read flow-through mode. When in flow-through mode, the 21554 can sustain long read bursts up to a 4KB aligned address boundary, or up to a page address boundary for upstream transactions falling into Memory Range 2. If the read data queue empties while the 21554 is in flow-through mode, the 21554 waits up to seven cycles and then disconnects if read data is still not available.

Prefetching

The 21554 prefetches read data to the aligned address boundaries shown in Table 4-3.

Table 4-3. Prefetch Boundaries

Read Command	Non-prefetchable Range	Prefetchable Range	In Flow-Through Mode
Memory Read	1 Dword	1 cache line	Page boundary for transactions in Upstream Memory 2 range 4KB boundary Initiator deasserts FRAME#
Memory Read Line	1 cache line	1 cache line	Page boundary for transactions in Upstream Memory 2 range 4 KB boundary Initiator deasserts FRAME#
Memory Read Multiple	2 cache lines	2 cache lines	Page boundary for transactions in Upstream Memory 2 range 4 KB boundary Initiator deasserts FRAME#

The cache line size corresponding to the initiator bus is used for determining prefetch boundaries.

Read Queue Full Threshold Tuning

The 21554 implements read queue management control bits for each read data queue in the Chip Control 1 configuration register. These bits specify at what read-queue threshold the 21554 initiates a delayed prefetchable read transaction on the target bus. Use of these bits can minimize fragmentation of prefetchable read bursts. The encoding and behavior of these bits are as follows:

- 00b: at least eight Dwords free in read data queue for all memory read commands
- 01b: at least eight Dwords free for all memory read commands (same as 00b)
- 10b: at least one cache line free for MRL and MRM, eight Dwords free for memory read
- 11b: at least one cache line free for all memory read commands

In these cases, the initiator bus cache line size is used. If the cache line size is not set to a valid value, then 8 Dwords is used for the read queue threshold.

4.1.4 Target Terminations

This section describes target retries, target disconnects, and target aborts received and returned by the 21554.

4.1.4.1 Target Terminations Returned by the 21554

The 21554 returns a target retry under the following circumstances:

- Queue is full for posted memory writes.
- Delayed transaction is queued but response is not ready.
- Queue is full for delayed transactions (delayed transaction not queued).
- Serial preload is ongoing.
- Primary Lockout Bit is set.
- Transaction is in progress for CSR generation of I/O or Configuration Access (delayed transaction not ready).

The 21554 returns a target disconnect under the following circumstances:

- Queue fills during posted write.
- Cache line boundary is reached for MWI transaction and the 21554 cannot buffer another.
- Cache line boundary is reached for memory write transaction and the Memory Write Disconnect bit is set.
- The 21554 runs out of read data during completion of delayed transaction to the initiator.
- Read transaction is nonprefetchable and multiple data phases are requested by the initiator.
- Multiple data phases requested by the initiator for an I/O or configuration access.
- Low two address bits are non-zero.

The 21554 returns a target abort and sets the Signaled Target Abort bit in the Primary or Secondary Status register under the following circumstances:

- Target abort is detected during a delayed transaction completion on the target bus.
- Master abort is detected in response to a delayed transaction on the target bus when the Master Abort Mode bit is set to a 1.
- Delayed transaction request is discarded after 2^{24} target retries received from the target.
- Invalid lookup table entry is encountered when forwarding upstream transactions in Upstream Memory 2 range and the Master Abort Mode bit is set to a 1.

4.1.4.2 Transaction Termination Errors on the Target Bus

When the 21554 detects a target abort on the target bus, the 21554 sets the Target Abort Received in the Primary or Secondary Status configuration register. In addition, the 21554 does the following:

- For delayed transactions, returns a target abort to the initiator and sets the Signaled Target Abort bit in the Primary or Secondary Status configuration register
- For posted write transactions, asserts SERR# on the initiator bus if the SERR# Enable for that interface is set, and sets the Signaled System Error bit in the Primary or Secondary Status configuration register

When the 21554 detects a master abort on the target bus, the 21554 always sets Master Abort Received bit in Primary or Secondary Status configuration register. In addition, the 21554 does the following:

- For delayed transactions when the Master Abort Mode bit is 0, returns TRDY# and, for reads, FFFFFFFFh to the initiator
- For delayed transactions when the Master Abort Mode bit is 1, returns a target abort and sets the Signaled Target Abort bit in the Primary or Secondary Status configuration register
- For posted write transactions, if SERR# Enable is set on the initiator bus interface and if SERR# Disable for Master Abort during Posted Write is clear, then the Signaled System Error bit is set in the Primary or Secondary Status register and SERR# is asserted on the initiator bus.

4.1.5 Ordering Rules

The 21554 can queue and forward multiple transactions at once. Therefore, at any one time the 21554 may have multiple posted write and multiple delayed transaction requests and completions queued and traveling in the same and opposite directions. The 21554 uses a set of ordering rules to dictate the order in which it initiates posted writes, initiates delayed transaction requests, and returns delayed transaction completion status. These rules reflect both the ordering constraints outlined in the *PCI Local Bus Specification, Revision 2.1* as well as implementation choices specific to the 21554.

Independent transactions on the primary and secondary buses only have a relationship when those transactions cross the 21554. General ordering guidelines for transactions crossing the 21554 are:

- The ordering relationship of a transaction with respect to other transactions is determined when the transaction completes; that is, when a transaction ends with a termination other than target retry.
- Requests terminated with target retry may be accepted and completed in any order with respect to other transactions that have been terminated with target retry. If the order of completion of delayed requests is important, the initiator should not start a second delayed transaction until the first one has been completed.
- Write transactions flowing in one direction have no ordering requirements with respect to write transactions flowing in the other direction. The 21554 can accept posted writes on both interfaces at the same time, as well as initiate posted writes on both interfaces at the same time.
- The acceptance of a posted memory write as a target can never be contingent on the completion of a non-posted transaction as a master. This is true of the 21554 and must also be true of other bus agents; otherwise, a deadlock can occur.
- The 21554 accepts posted writes regardless of the state of completion of any delayed transactions being forwarded across the bridge.
- A target retry in response to a posted write is allowed, but only due to temporary conditions, such as a buffer-full condition.

The ordering rules apply to posted write, delayed write and read request, and delayed write and read completion transactions crossing the bridge in the same direction. Note that delayed completions cross the bridge in the opposite direction of its respective delayed request. Table 4-4 shows the 21554 transaction ordering rules.

Table 4-4. 21554 Transaction Ordering Rules

↓ Pass→	Posted Write	Delayed Read Request	Delayed Write Request	Delayed Read Completion	Delayed Write Completion
Posted Write	No	Yes	Yes	Yes	Yes
Delayed Read Request	No	Yes	Yes	Yes	Yes
Delayed Write Request	No	Yes	Yes	Yes	Yes
Delayed Read Completion	No	Yes	Yes	Yes	Yes
Delayed Write Completion	Yes	Yes	Yes	Yes	Yes

The only ordering restriction the 21554 enforces is ordering with respect to posted writes. No other transaction other than a delayed write completion can pass a posted write. Posted writes are delivered in the order in which they are accepted.

Delayed transactions may be initiated by the 21554 in any order, and are not necessarily initiated in the order in which they are received. When the 21554 initiates a delayed transaction, the 21554 can behave in one of two ways. If the Delayed Transaction Ordering Control configuration bit is not set, the 21554 uses a rotating fairness algorithm to select which delayed transaction it initiates next, regardless of the type of target termination is returned (retry, TRDY#, etc.). If the Delayed Transaction Ordering Control configuration bit is set, the 21554 continues to initiate the same transaction until a response other than target retry is received.

Delayed completions are returned to the initiator when ready, regardless of the order in which corresponding delayed requests were queued. A delayed read completion may not be returned to the initiator (the initiator receives a target retry) if a posted write is ahead of the delayed completion in the queues. That is, the write was posted in the direction of the completion, but before the read data was queued. In this case the write must be delivered before the read data can be returned to the initiator.

Address Decoding 5

This chapter presents information about address decoding.

The 21554 supports independent primary and secondary address maps. The 21554 implements separate base address registers (BARs) on both the primary and secondary interfaces that denote address ranges for downstream and upstream forwarding, respectively. The 21554 also has primary and secondary BARs denoting ranges for CSR access. On the primary interface, the 21554 only responds to those transactions whose addresses fall into one of its primary BAR ranges. All other I/O and memory transactions on the primary bus are ignored by the 21554. Similarly, on the secondary interface, the 21554 responds only to those transactions whose addresses reside in one of the secondary BAR ranges. All other transactions on the secondary bus are ignored by the 21554.

The address ranges defined by the primary BARs reside in the primary, or system, address map. The address ranges defined by the secondary BARs reside in the secondary, or local, address map. The 21554 supports address translations between the two address maps when forwarding transactions upstream or downstream.

As a target, the 21554 ignores any transactions that it initiates as a master.

5.1 21554 CSR Address Decoding

The 21554 implements a set of control and status registers that may be mapped in either memory or I/O space. The registers are mapped independently on the primary and secondary interfaces. The following BARs are used to map the 21554 CSRs:

- Primary CSR and Downstream Memory 0 BAR for mapping in primary bus memory address space
  - — Lower 4KB of this range used to map the 21554 CSRs
- Primary CSR I/O BAR for mapping in primary bus I/O space
- Secondary CSR Memory BAR for mapping in secondary bus memory address space
- Secondary CSR I/O BAR for mapping in secondary bus I/O space

The primary BARs are located in the 21554 primary bus configuration space, and the secondary BARs are located in the 21554 secondary bus configuration space. The memory BARs request 4 KB each (minimum size for Primary CSR and Downstream Memory 0 BAR), and the I/O BARs request 256 bytes each.

5.2 Expansion ROM Address Decoding

The 21554 implements one BAR, the Primary Expansion ROM BAR, to map the expansion ROM that may be attached to the 21554. The Expansion ROM can be mapped into primary bus address space only, and is not accessible through a BAR from the secondary bus. The size of the Primary Expansion ROM BAR is programmable through the Primary Expansion ROM Setup register in device-specific configuration space. The size may vary from 4 KB to 16 MB by powers of 2. The Primary Expansion ROM BAR may also be disabled through the setup register so that it requests no space if the expansion ROM is not implemented.

5.3 Memory Transaction Address Decoding

The 21554 implements four BARs on the primary interface and three BARs on the secondary interface that may be enabled to decode and forward memory transactions to the opposite interface. These BARs are:

- Primary CSR and Downstream Memory 0 BAR, for addresses above the low 4 KB in this address range
- Downstream I/O or Memory 1 BAR
- Downstream Memory 2 BAR
- Downstream Memory 3 BAR
- Upstream I/O or Memory 0 BAR
- Upstream Memory 1 BAR
- Upstream Memory 2 BAR

The downstream BARs are located in primary configuration space and are used to decode transactions on the primary bus to be forwarded to the secondary bus. The upstream BARs are located in secondary configuration space and are used to decode transactions on the secondary bus to be forwarded to the primary bus.

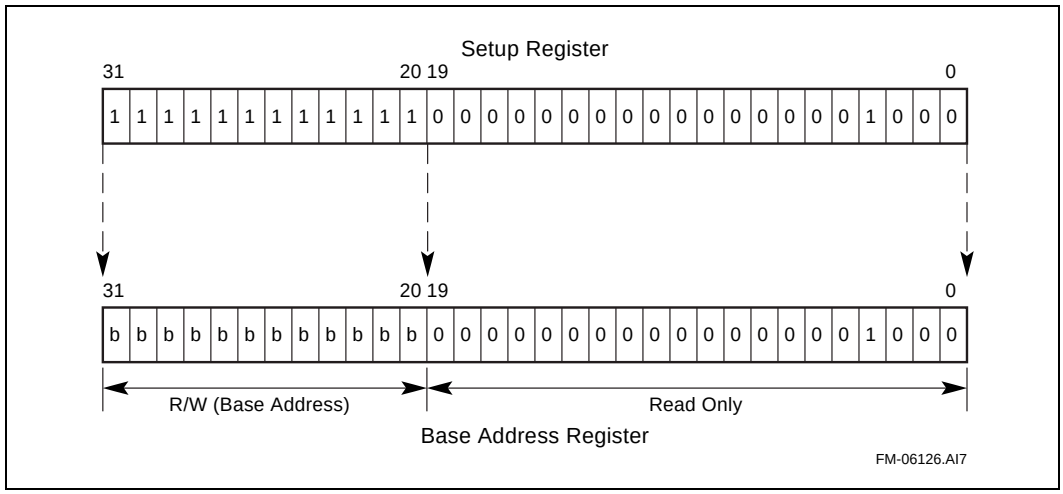
5.3.1 Using the BAR Setup Registers

All downstream and upstream BARs have programmable sizes, and may be disabled so that they request no space. The Primary CSR and Downstream Memory 0 BAR cannot be totally disabled, as the 21554 CSRs are always mapped in the bottom 4KB. The forwarding part of the range may be disabled by requesting only 4KB of memory. (Table 5-2 summarizes the minimum and maximum range for each address range.) In addition, the Downstream Memory 3 BAR can be configured to be mapped in 64-bit address space. The register then comprises two 32-bit registers and can be used for forwarding DACs downstream. 64-bit addressing support is discussed further in Section 5.3.4. These BARs can also be programmed to be prefetchable or non-prefetchable.

Programming of all the forwarding BARs with the exception of the Upstream Memory 2 BAR is done through corresponding device-specific setup configuration registers. The Primary Expansion ROM BAR also has a setup register. Setup registers are preloaded by the serial ROM and are also writeable from the secondary interface. Each bit of the setup register corresponds to the same bit of its respective BAR. Bit 0 of the Downstream I/O or Memory 1 Setup register and the Upstream I/O or Memory 0 Setup register should be written with a 0 to select a memory BAR or a 1 to select an I/O BAR. Bits [2:1] are writeable to select the type of memory mapping. The Downstream Memory 3 Setup registers bits [2:1] may be set to 10b to select 64-bit addressing.

A mask is used to set the size of the BAR for the remaining read/write bits of the setup register. Writing a 1 sets the corresponding bit in that setup register's BAR to be read/write. Writing a 0 sets the corresponding bit in that setup register's BAR to be read only as 0. Therefore, the size is set by writing the appropriate number of most significant bits to a 1, and the remaining bits to a 0. If all the zeros and ones in the size field are not contiguous, this is illegal and unpredictable results may occur. If the most significant writeable bit of the setup register is a 0, then the corresponding BAR is disabled and requests no space. Figure 5-1 shows an example of using a setup register to program a BAR to request 1 MB of memory space.

Figure 5-1. BAR Setup Register Example

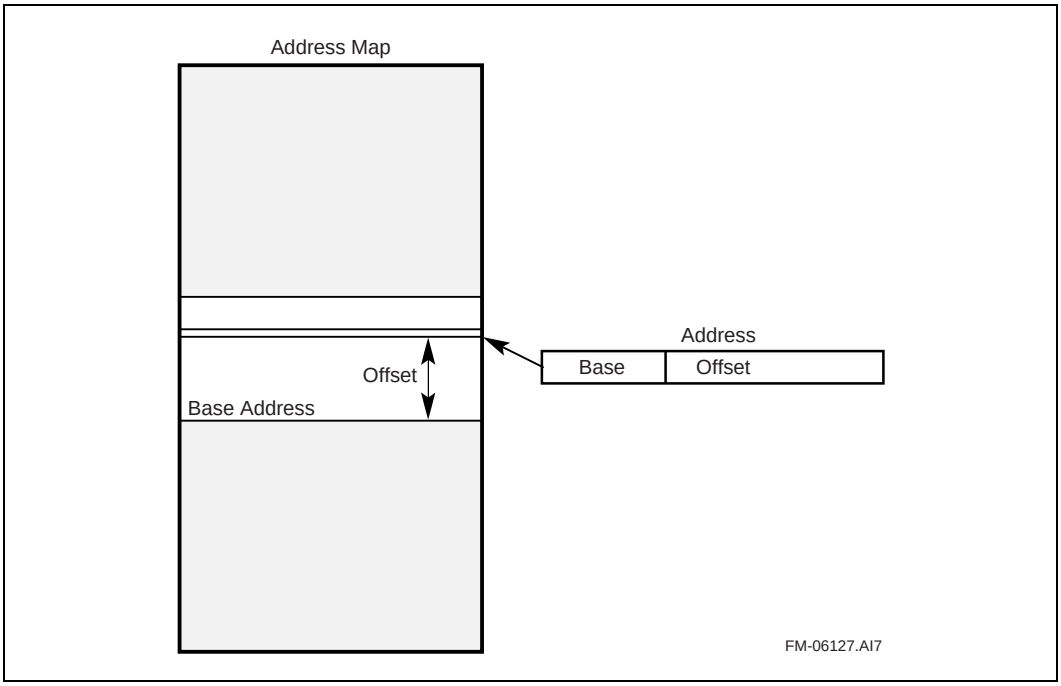


5.3.2 Direct Address Translation

With the exception of secondary bus transactions falling into the Upstream Memory 2 address range and all dual address transactions, the 21554 uses direct address translation when forwarding memory transactions from one interface to the other. Note that since transactions addressing the bottom 4 KB of the Primary CSR and Downstream Memory 0 BAR are targeted at the 21554 CSRs, no forwarding and therefore no address translation is performed. Direct address translation is used for transactions in that range above the low 4 KB boundary.

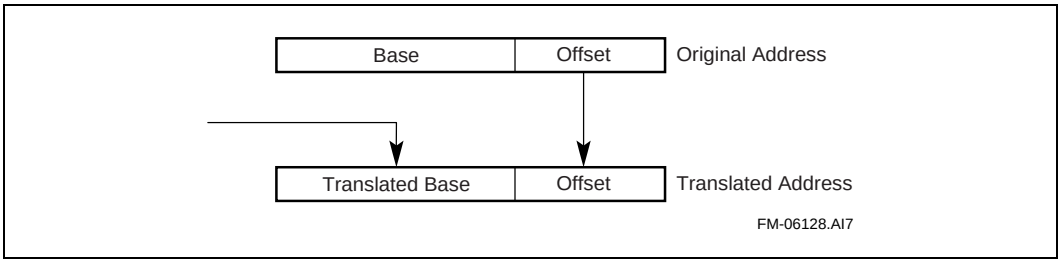
A memory address may be thought of as a base address (as programmed in the Downstream and Upstream BARs) with an offset from the base address, as shown in Figure 5-2.

Figure 5-2. Address Format



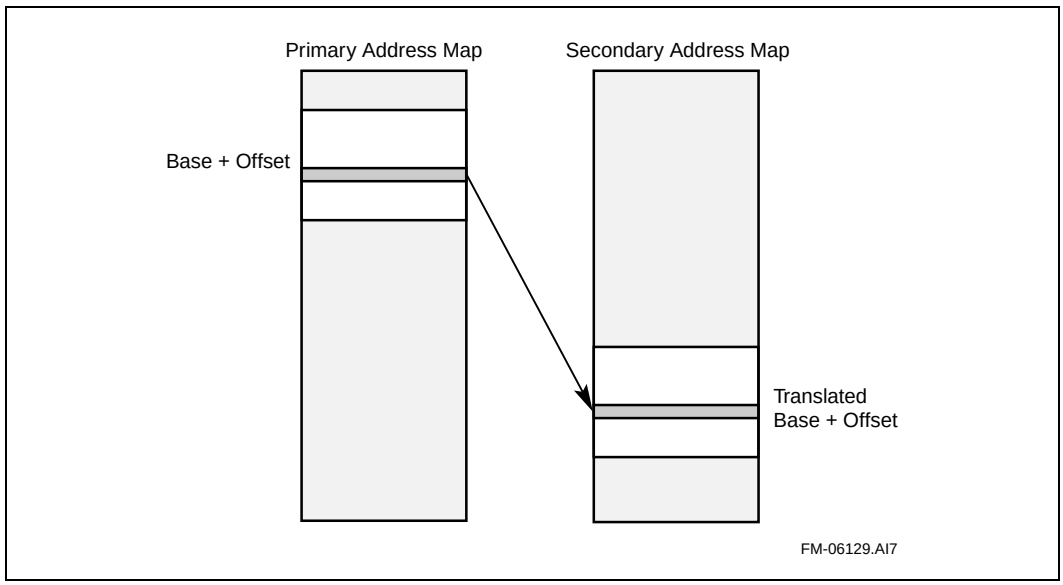
When a memory transaction is forwarded downstream from the primary bus to the secondary bus, the primary bus address may be mapped to another address in the secondary bus domain by substituting a new base address for the base of the original address, as shown in Figure 5-3. This new base address, also called the translated base address, references a new location in the secondary bus address map. The offset is not affected. The process is similar for transactions forwarded from the secondary bus to the primary bus.

Figure 5-3. Direct Offset Address Translation



Each memory address range using direct offset address translation has its own translated base. The translated base addresses are programmable in registers corresponding to each BAR. These registers are mapped both in device-specific configuration space and in CSR space. The number of bits of the translated base address corresponds to the number of writeable bits in the respective BAR. Likewise, the number of bits of the offset also varies and depends on the size of the BAR. Figure 5-4 gives an example of address translation of downstream memory transactions. Again, upstream transactions are treated similarly.

Figure 5-4. Downstream Address Translation Example



5.3.3 Lookup Table Based Address Translation

As mentioned previously, Upstream Memory 2 address translation is treated differently than the other ranges. The 21554 uses a page size based lookup table to perform address translation for transactions falling into this range. A lookup table provides a flexible way of translating secondary bus local addresses to primary bus system addresses.

The Upstream Memory 2 address range consists of a fixed number (64) of pages. The page size is programmable in the Chip Control 1 configuration register. Therefore, the size of the Upstream Memory 2 BAR is dependent on the page size. The page size varies between 256 bytes to 4 MB by powers of 2. This results in a window size that varies from 16 KB to 256 MB. This BAR can also be disabled.

Each page of the upstream window has a corresponding translated base address.

The size of the translated base address varies with the page size and window size. The translated base address replaces both the original base address and the lookup table index bits. The address bits used for the original base address for a given page and window size are shown in Table 5-1. Also shown are the locations of the six address bits needed to select one of the 64 entries in the lookup table. Finally, the offset of the address, which is not translated, consists of the remaining lower order address bits. Table 5-1 shows the Upstream Memory 2 window size, with base address, index, and offset fields.

Table 5-1. Upstream Memory 2 Window Size

Page Size (bytes)	Window Size (bytes)	Base Address (bits)	Lookup Table Index (bits)	Offset (bits)
256	16K	[31:14]	[13:8]	[7:0]
512	32K	[31:15]	[14:9]	[8:0]
1K	64K	[31:16]	[15:10]	[9:0]
2K	128K	[31:17]	[16:11]	[10:0]
4K	256K	[31:18]	[17:12]	[11:0]
8K	512K	[31:19]	[18:13]	[12:0]
16K	1M	[31:20]	[19:14]	[13:0]
32K	2M	[31:21]	[20:15]	[14:0]
64K	4M	[31:22]	[21:16]	[15:0]
128K	8M	[31:23]	[22:17]	[16:0]
256K	16M	[31:24]	[23:18]	[17:0]
512K	32M	[31:25]	[24:19]	[18:0]
1M	64M	[31:26]	[25:20]	[19:0]
2M	128M	[31:27]	[26:21]	[20:0]
4M	256M	[31:28]	[27:22]	[21:0]

Figure 5-5 shows how a translated address is built using the lookup table, assuming a page size of 4 KB.

Figure 5-5. Address Translation Using Lookup Table

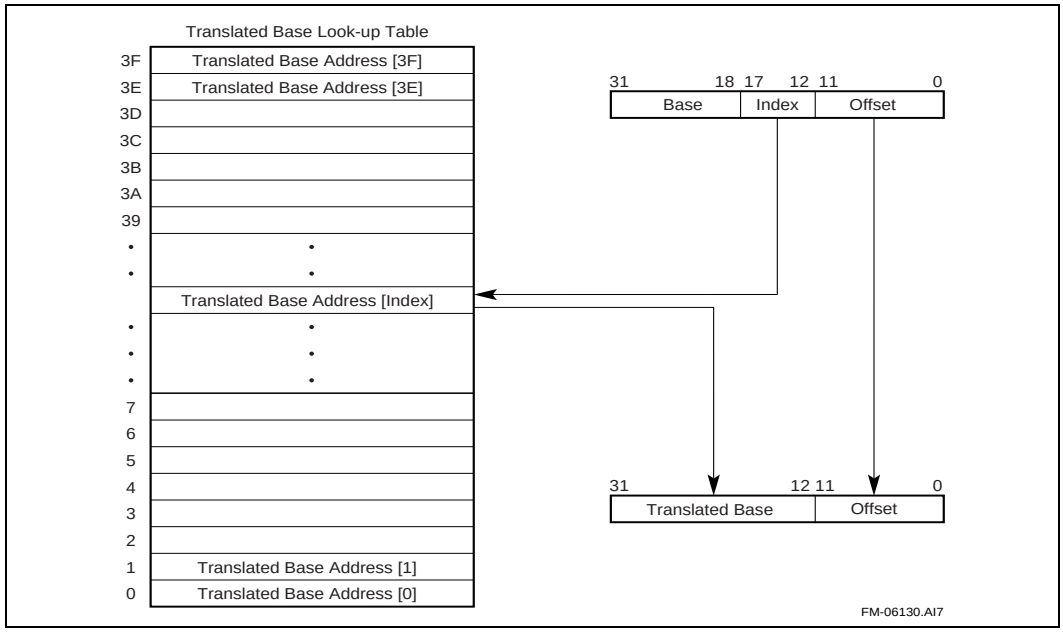
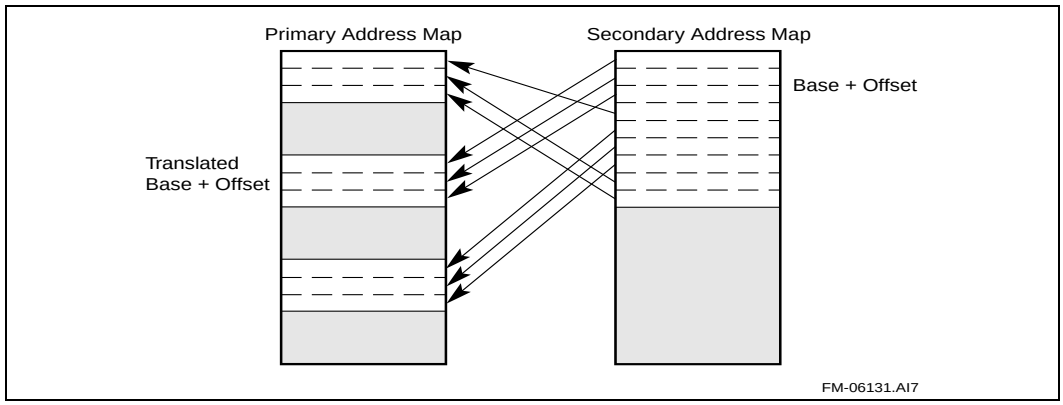


Figure 5-6 shows an example of how different address regions might be forwarded upstream using the lookup table address translation.

The lookup table is part of the memory space that the 21554 requests with its Primary CSR Memory BAR and Secondary CSR Memory BAR. The lookup table is indirectly accessible in I/O or memory space at offsets 24h and 28h. This table is implemented on-silicon. No external memory is needed.

Figure 5-6. Upstream Lookup Table Address Translation



The 21554 conditionally asserts s_inta_l when an upstream memory transaction transfers data addressing the last Dword in a page. This interrupt alerts the local processor that the page entry may need updating. The 21554 implements an event bit and interrupt mask bit for each of the 64 pages (entries) in the upstream window.

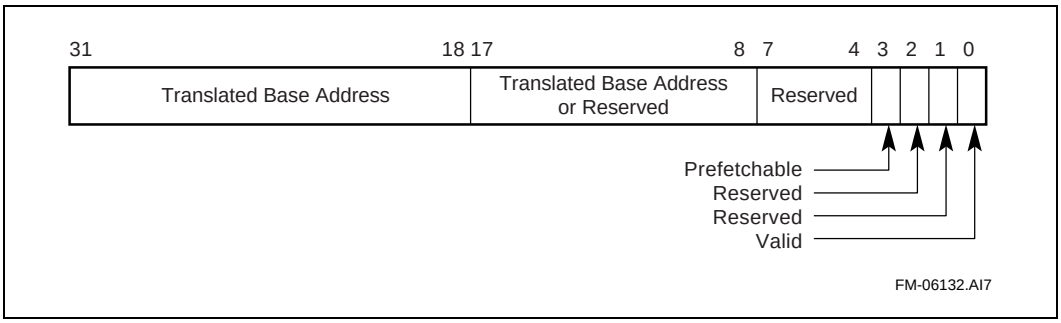
Note: The page entry of the lookup table should not be updated while the initiator is still performing transactions addressing that page.

5.3.3.1 Lookup Table Entry Format

Figure 5-7 shows the format for an entry in the lookup table. The number of bits of the entry used for the new translated base address is variable. The maximum number of bits used are bits [31:8], corresponding to a 256-byte page size, while the minimum number of bits used are bits [31:22], corresponding to a 4 MB page size. The next 4 to 18 bits, depending on the number of bits used for the base address, are reserved. The low 4 bits are used for control. Two control bits are defined, one indicating whether the entry is a valid entry, and one indicating whether prefetchable behavior should be used on memory reads. If the entry is not valid, then the 21554 treats the transaction addressing that page as if a master abort were detected on the target interface. For writes, the 21554 discards memory write data and, if the Secondary SERR# Disable for Master Abort during Posted Write bit is 0 and the SERR# Enable bit is 1, asserts s_serr_l. For reads, the 21554 returns FFFFFFFFh on reads if the Master Abort Mode bit is 0, or returns a target abort if the Master Abort Mode bit is a 1.

Note: The lookup table is not cleared by reset. The lookup table must be initialized by the local processor before the Upstream Memory 2 Address range is used.

Figure 5-7. Lookup Table Entry Format



5.3.4 Forwarding of 64-Bit Address Memory Transactions

The 21554 considers the host and local memory space above the 4 GB boundary to be shared. This means that the 21554 uses a flat address map in this space. Dual-address cycle (DAC) transactions are used for addressing above the 4 GB boundary. The 21554 can forward dual-address cycle transactions both upstream and downstream. The Downstream Memory 3 BAR is used to designate the address range for downstream DACs. Inverse decoding is used for upstream DACs.

The Downstream Memory 3 BAR may be configured to be a 64-bit BAR by preloading the Downstream Memory 3 Upper 32 Bits Setup register bit [31] to a one. The Downstream Memory 3 Setup register bits [2:1] should be set to 10b. This implies that the memory range can be located anywhere in 64-bit address space. If this 64-bit addressing option is used, the maximum window size changes from 2 GB (in the 32-bit case) to 2^{63} bytes.

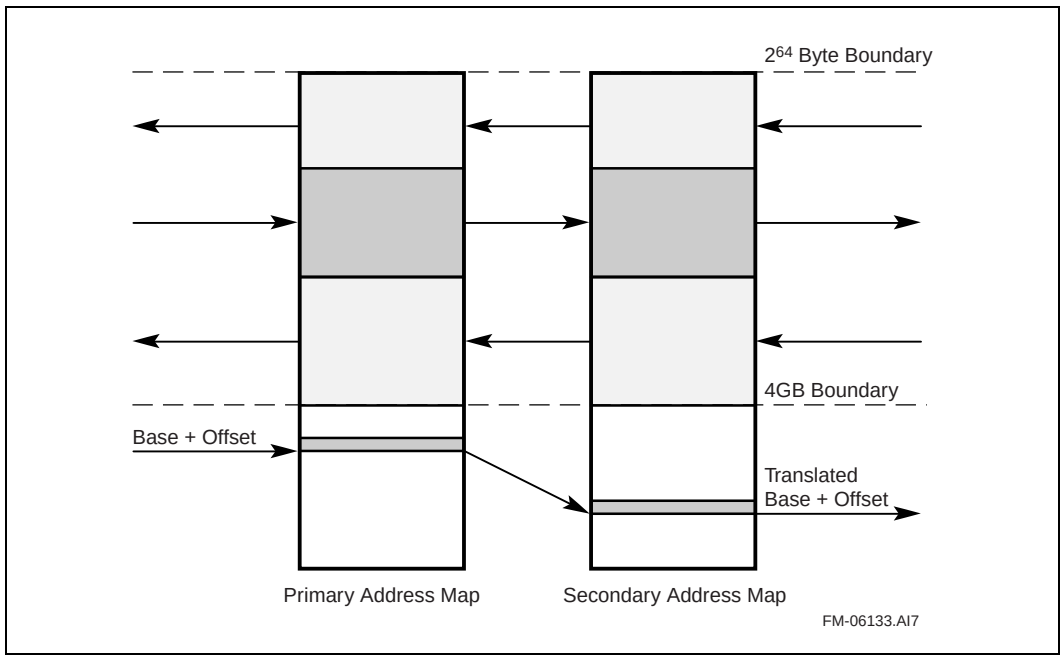
If the preloaded window size for a 64-bit BAR is 2 GB or less, then the space requested may be mapped either in 32-bit address space or 64-bit address space. In the former case, the upper 32 bits of the base address is zero and transactions are forwarded as described in the previous section using direct offset address translation. If the upper 32-bit base address is non-zero, then the memory range is located above the 4 GB boundary.

If the Downstream Memory 3 Range is mapped above the 4 GB boundary, then primary bus transactions falling into this address range are forwarded downstream with no address translation performed. Any 64-bit address transactions on the secondary bus falling outside of the

Downstream Memory 3 address range are forwarded upstream, again with no address translation. This is similar to the forwarding mechanisms of a standard PCI-to-PCI bridge and is illustrated in Figure 5-8.

Note that since the use of BARs restricts the alignment of the address range to the window size, the Downstream Memory 3 address range can never straddle the 4 GB boundary.

Figure 5-8. Dual-Address Transaction Forwarding



5.4 I/O Transaction Address Decoding

The 21554 provides a mechanism where one BAR on each interface can be configured to be an I/O BAR instead of a memory BAR. The Downstream I/O or Memory 1 BAR in primary configuration space is used to decode primary bus I/O transactions for forwarding to the secondary bus. The Upstream I/O or Memory 0 BAR in secondary configuration space is used to decode secondary bus I/O transactions for forwarding to the primary bus.

The 21554 performs direct offset address translation when forwarding I/O transactions in much the same manner that it translates memory addresses. The size of the I/O BARs can be configured to be 64 bytes, 128 bytes, or 256 bytes. Accordingly, the base address can consist of 26, 25, or 24 bits. The 21554 hardware does not restrict setting up larger I/O windows, although requesting more than 256 bytes of I/O space is a violation of the *PCI Local Bus Specification, Revision 2.1.* The upper bits comprising the base address of the I/O address on the primary bus is replaced with the base address written in the Downstream I/O or Memory 1 Translated BAR when initiated on the secondary bus. Similarly, the Upstream I/O or Memory 0 Translated BAR is used for upstream I/O transactions. These translated base registers are mapped in both device-specific configuration space and the 21554 CSR space.

5.4.1 Indirect I/O Transaction Generation

The 21554 implements a CSR mechanism that allows access to any I/O address in the secondary or local I/O address map from the primary interface, or any I/O address in the primary or host I/O address map from the secondary interface. A pair of device-specific CSR registers contain the address and data used to initiate the I/O transaction. One pair is used for downstream I/O transactions and one pair is used for upstream I/O transactions. The downstream registers can only be accessed from the primary interface, and the upstream registers can only be accessed from the secondary interface. They function similarly, so only the downstream case is discussed.

The Downstream I/O Address register contains the address used when the transaction is initiated on the secondary bus. When the Downstream I/O Data register is read or written from the primary interface, the 21554 initiates the transaction on the secondary bus. For writes, the Downstream I/O Data register contains the write data to be written. For reads, the read data is placed in this register upon completion of the secondary bus I/O read.

The I/O Data register must be accessed with an I/O transaction on the primary interface to initiate the secondary bus I/O transaction. Otherwise, this register appears as reserved for both memory accesses or accesses from the secondary interface. In addition, the Downstream I/O Control bit in the I/O Control and Status register must be set to enable downstream I/O transaction generation; otherwise, I/O Data register accesses are treated as reserved accesses.

The 21554 uses the same byte enables that the initiator used to read or write the register.

Note: The low bits of the I/O address written in the I/O Address register must match the byte enables used during the data phase, as described in the *PCI Local Bus Specification, Revision 2.1*. The 21554 will not correct discrepancies between byte enables and address bits [1:0].

The 21554 responds to read or write access of Downstream I/O Data register with a target retry until the I/O transaction is completed on the secondary bus. This I/O access is treated as a delayed transaction by the 21554. This delayed transaction is entered into the 21554's downstream delayed transaction queue and is ordered with respect to all other downstream transactions. When ordering rules permit, the 21554 initiates I/O write or read on the secondary bus. When the I/O transaction completes, then the 21554 returns target termination and, if a read, returns read data when the initiator repeats the transaction.

The 21554 provides a semaphore method that may be used to guarantee atomicity of the Downstream I/O Address and Downstream I/O Data register accesses using the Downstream I/O Own bit. Atomicity of these accesses is not hardware-enforced. An Upstream I/O Own bit is provided for upstream I/O transactions. The following procedure should be used for downstream I/O transactions:

1. The initiator of the transaction reads the Downstream I/O Own bit. If the bit reads as zero, then the initiator may proceed with the indirect I/O transaction sequence. If the bit reads as a 1, then the initiator should not proceed until a subsequent read of the own bit returns a 0. The 21554 automatically sets the own bit to a 1 after it is read from the primary interface.
2. The initiator writes the target I/O address in the Downstream I/O Address register.
3. The initiator should write or read the data in the Downstream I/O Data register until a response other than target retry is received.
4. Upon returning the completion of the I/O transaction to the initiator, the 21554 automatically clears the bit to a 0.

The same procedure should be used for upstream I/O transactions using the Upstream I/O Address register, Upstream I/O Data register, and Upstream I/O Own bit. To read the state of the Downstream and Upstream I/O Own bits without side effects, a read-only copy of the I/O Own bit states is kept in the I/O Control and Status register. Byte access of the I/O Own bits and their read-only copies should be used to avoid setting the I/O Own bit for the opposite interface.

5.4.2 Subtractive Decoding of I/O Transactions

The 21554 can be enabled to subtractively decode I/O transactions and forward these transactions to the opposite bus. No address translation is performed on subtractively decoded I/O transactions. The transaction is treated by the 21554 as a delayed transaction. To enable subtractive decoding of I/O transactions on the primary bus, the Subtractive Decode Enable bits in the Chip Control 1 configuration register must be set to 01b. To enable subtractive decoding of I/O transactions on the secondary bus, the Subtractive Decode Enable bits must be set to 10b.

Note: There can be only one subtractive decoding agent on a PCI bus. Subtractive decoding should not be enabled for both the primary and secondary interfaces.

5.5 21554 Base Address Register Summary

Table 5-2 shows a summary of the 21554 BARs.

Table 5-2. Base Address Register Summary

Base Address Register	Size	Address Translation
Primary CSR and Downstream Memory 0 BAR	4 KB to 2 GB	Low 4 KB: None Above 4KB boundary: Direct Offset
Primary CSR I/O BAR	256 bytes	—
Secondary CSR Memory BAR	4 KB	—
Secondary CSR I/O BAR	256 bytes	—
Primary Expansion ROM BAR	4 KB to 16 MB	—
Downstream I/O or Memory 1 BAR	64 bytes to 256 bytes (I/O) or 4 KB to 2 GB (memory)	Direct Offset
Downstream Memory 2 BAR	4 KB to 2 GB	Direct Offset
Downstream Memory 3 BAR	4 KB to 2^{63} bytes	Direct Offset (< 4 GB) None (≥4 GB)
Upstream I/O or Memory 0 BAR	64 bytes to 256 bytes (I/O) or 4 KB to 2 GB (memory)	Direct Offset
Upstream Memory 1 BAR	4 KB to 2 GB	Direct Offset
Upstream Memory 2 BAR	16 KB to 256 MB	Lookup Table

Configuration Accesses 6

This chapter describes how the 21554 responds to Type 0 configuration transactions.

The 21554 implements two sets of configuration registers: one for the primary interface and one for the secondary interface. Both sets are accessible from either interface. The 21554 can act as an initiator of Type 0 or Type 1 configuration transactions on the primary or secondary bus using the indirect configuration transaction mechanism.

As a target, the 21554 ignores any transactions that it initiates as a master.

6.1 Type 0 Accesses to the 21554 Configuration Space

The 21554 responds as a target to Type 0 configuration transactions on both its primary and secondary interfaces when the IDSEL pin for that interface is asserted. The 21554 is a single-function device and does not decode the function number. Because the 21554 is not a transparent PCI-to-PCI bridge, it does not respond to Type 1 configuration transactions.

Access to the 21554 configuration space may be restricted during different phases of initialization:

- Reset:

  No access to the 21554 configuration space from either interface.

- Serial preload:

  No access to the 21554 configuration space from either interface. The 21554 returns target retry.

- Optional primary lockout:

  Access to the 21554 configuration space is allowed from the secondary interface only, until the Primary Lockout Bit in the Chip Control 0 register is cleared. The 21554 returns target retry to all accesses initiated on the primary bus, with the exception of accesses to the Reset Control register at Dword D8h.

- Normal configuration and operation:

  Access to the 21554 configuration space is allowed from both the primary and secondary interfaces.

See Chapter 16 for a more detailed description of the initialization process.

Accesses to the 21554 configuration space are not ordered with respect to transactions in the 21554 queues. That is, the 21554 responds immediately to configuration transactions regardless of what transactions exist in the upstream and downstream queues. Exceptions to this are configuration accesses that result in the initiation of configuration and I/O transactions by the 21554. These transactions are entered in the delayed transaction queue and ordered appropriately with respect to other delayed transactions and posted writes in the 21554 queues.

6.2 Initiation of Configuration Transactions by 21554

Usually, the host processor configures primary bus devices and the local processor configures secondary bus devices, so forwarding of configuration transactions across the 21554 is typically not necessary. However, in order to support other configuration methods, the 21554 implements a mechanism that enables initiation of Type 0 or Type 1 configuration accesses on either the primary bus or the secondary bus. This mechanism is different from the hierarchical mechanisms supported by PCI-to-PCI bridges. Instead, two pairs of device-specific registers contain the address and data that are used to initiate the configuration transaction. One pair is used to generate transactions on the primary interface; the other is used to generate transactions on the secondary interface:

- The Upstream Configuration Address and Upstream Configuration Data registers contain the address and data of the configuration transaction to be initiated on the primary bus.
- The Downstream Configuration Address and Downstream Configuration Data registers contain the address and data of the configuration transaction to be initiated on the secondary bus.

In addition, the Configuration Control and Status register and Configuration Own Bits register are used for configuration transaction generation. All of these registers are mapped into both device-specific configuration space and the 21554 CSR space. The upstream address and data registers can be written from the secondary interface only, and the downstream address and data registers can be written from the primary interface only. Downstream and upstream configuration address registers can be read from either interface.

In order to generate a configuration transaction, the corresponding Upstream or Downstream Configuration Control bit in the Configuration Control and Status register must be set. Otherwise, the corresponding Data registers are treated as reserved registers. The Configuration Data registers are also treated as reserved registers in memory space.

The Upstream or Downstream Configuration Address register must be written with the address to be driven before the corresponding data register is accessed. This address is driven on the AD lines exactly as written in the register. Therefore, a Type 0 format must be used to generate a Type 0 configuration transaction, and a Type 1 format must be used to generate a Type 1 configuration transaction. The upper 21 bits of a Type 0 address format are used as IDSEL signals and are specific to the motherboard or add-in card application.

The configuration transaction is initiated by the 21554 when the Upstream or Downstream Configuration Data register is either read or written from the secondary or primary interface, respectively. These registers must be accessed by either a configuration transaction or an I/O transaction to initiate the transaction. The 21554 uses the same byte enables that the initiator used to read or write the register. The 21554 responds to the access of the Upstream or Downstream Configuration Data register with a target retry until the access is completed on the target bus. When the access is completed, the 21554 returns the corresponding target termination and, if a read, the read data on a subsequent attempt of the transaction by the initiator. If the Delayed Transaction Target Retry Counter expires, that is, 2^{24} target retries are received from the target, then the 21554 returns a target abort to the initiator.

The 21554 provides a semaphore method that may be used to guarantee atomicity of the address and data register accesses using the Upstream Configuration Own bit and Downstream Configuration Own bit. Atomicity of these accesses is not guaranteed in hardware. If the corresponding Configuration Control bit is not set, then the Own bit is treated as reserved. The following procedure should be used for downstream transactions:

1. The initiator of the transaction should read the Downstream Configuration Own bit for initiation of transactions on the secondary bus. If the bit reads as zero, then the initiator may proceed with the configuration transaction sequence. If the bit reads as a 1, then the initiator should not proceed until a subsequent read of the own bit returns a 0. The 21554 automatically sets the own bit to a 1 after it is read.
2. The initiator should write the target configuration address in the Downstream Configuration Address register.
3. The initiator should write or read the data in the Downstream Configuration Data register until a response other than target retry is received.
4. Upon completion of the configuration transaction on the initiator bus, the 21554 automatically clears the Downstream Configuration Own bit to a 0.

Upstream configuration transactions should use a similar process. To check the status of the own bits without read side effects, read-only copies of these bits are located in the Configuration Control and Status register. Byte access of the Configuration Own bits and their read-only copies should be used to avoid setting the Configuration Own bit for the opposite interface.

Configuration Space Registers 7

This chapter describes the 21554 configuration space registers.

The 21554 configuration space is divided into three parts: the primary interface configuration registers, the secondary interface configuration registers, and the device-specific configuration registers. Both the primary and secondary interface configuration headers contain the 64-byte Type 0 configuration header corresponding to that interface. The device-specific configuration registers are specific to the 21554, some of which apply to the primary interface, others to the secondary interface, and some to other 21554 functions.

The 21554 provides very flexible configuration mechanisms. Access to the 21554 configuration registers is supported from both the primary and secondary interfaces. Some configuration parameters can be preloaded from the serial ROM attached to the 21554 prior to initialization. This enables vendor-specific configuration parameters to be loaded into the 21554 configuration registers, replacing default values specified by Intel. These vendor-specific parameters are loaded before configuration of the 21554 by the local and/or host processors. Parameters that can be preloaded include address mapping requirements, Class Code, Subsystem ID and Subsystem Vendor ID, and others. Section 7.3 contains a description of the preload sequence and format. During the preload operation, all accesses to the 21554 configuration registers receive a target retry.

Subsequent to the serial ROM preload, the local processor may then access configuration registers before the host processor is allowed access. The Primary Lockout configuration space bit can be set to prevent access by the primary interface. The local processor can control this bit. When the Primary Lockout is set, the 21554 returns a target retry to any configuration accesses by the host processor, with the exception of the Reset Control register at Dword location D8h. When the bit is cleared, the 21554 is then able to return TRDY# in response to host configuration accesses. Optionally, the Primary Lockout bit may be cleared during the serial preload, permitting simultaneous host and local access to configuration space. The Reset Control register is always accessible to the host, except during chip reset.

Read accesses to reserved or unimplemented registers complete normally and return a data value of zero when read. Writes to reserved registers are completed normally and the write data discarded.

Software must be careful when accessing registers that have bit fields reserved for future use. For read accesses, software must use appropriate masks to extract the defined bits and may not rely on reserved bits being any particular value. For write accesses, software must ensure that the values of reserved bit positions are preserved. That is, the values of the reserved bit positions must first be read, merged with the new values for other bit positions, and the merged data then written back.

7.1 Configuration Space Address Map

Figure 7-1 shows an address map of the 21554 primary interface configuration space. Figure 7-2 shows an address map of the 21554 secondary interface configuration space. Figure 7-3 shows an address map of the 21554 device-specific configuration space.

The first 64 bytes of the primary address map contain the primary interface configuration registers. The first 64 bytes of the secondary address map contain the secondary interface configuration registers. Some of these registers are shared copies, such as the Device ID. Other registers are separate, such as the Primary Cache Line Size and the Secondary Cache Line Size. Separate registers have either "Primary" or "Secondary" preceding the register name. The secondary configuration registers can be accessed in the second 64 bytes of the primary interface's configuration space map (primary addresses 40h - 7Fh). Similarly, the primary configuration registers can be accessed in the second 64 bytes of the secondary interface's configuration space map (secondary addresses 40h - 7Fh). The remaining device-specific registers are accessible at the same configuration address from both the primary and secondary interfaces.

Figure 7-1. Primary Interface Configuration Space Address Map

Byte 3	Byte 2	Byte 1	Byte 0	Primary Offset	Secondary Offset
Device ID [1]		Vendor ID [1]		00h	40h
Primary Status		Primary Command		04h	44h
Primary Class Code [2]			RevID [1]	08h	48h
BiST [1,2]	Header Type [1]	Primary MLT	Primary CLS	0Ch	4Ch
Primary CSR and Downstream Memory 0 BAR				10h	50h
Primary CSR I/O BAR				14h	54h
Downstream I/O or Memory 1 BAR				18h	58h
Downstream Memory 2 BAR				1Ch	5Ch
Downstream Memory 3 BAR				20h	60h
Upper 32 Bits Downstream Memory 3 BAR				24h	64h
Reserved				28h	68h
Subsystem ID [1,2]		SubsystemVendor ID [1,2]		2Ch	6Ch
Primary Expansion ROM Base Address				30h	70h
Reserved			Capabilities Pointer [1]	34h	74h
Reserved				38h	78h
Primary MAX_LAT [2]	Primary MIN_GNT [2]	Primary Interrupt Pin	Primary Interrupt Line	3Ch	7Ch

1. Primary and secondary configuration registers are shared.
2. Register or a portion of the register may be preloaded using the serial ROM interface.

FM-06134.AI4

Figure 7-2. Secondary Interface Configuration Space Address Map

Byte 3	Byte 2	Byte 1	Byte 0	Primary Offset	Secondary Offset
Device ID[1]		Vendor ID[1]		40h	00h
Secondary Status		Secondary Command		44h	04h
Secondary Class Code[2]			RevID[1]	48h	08h
BiST[1,2]	Header Type[1]	Secondary MLT	Secondary CLS	4Ch	0Ch
Secondary CSR Memory BAR				50h	10h
Secondary CSR I/O BAR				54h	14h
Upstream I/O or Memory 0 BAR				58h	18h
Upstream Memory 1 BAR				5Ch	1Ch
Upstream Memory 2 BAR				60h	20h
Reserved				64h	24h
Reserved				68h	28h
Subsystem ID[1,2]		Subsystem Vendor ID[1,2]		6Ch	2Ch
Reserved				70h	30h
Reserved			Capabilities Pointer[1]	74h	34h
Reserved				78h	38h
Secondary MAX_LAT[2]	Secondary MIN_GNT[2]	Secondary Interrupt Pin	Secondary Interrupt Line	7Ch	3Ch

1. Primary and secondary configuration registers are shared.
2. Register or a portion of the register may be preloaded using the serial ROM interface.

FM-06135.AI4

Figure 7-3. Device-Specific Configuration Address Map

Byte 3	Byte 2	Byte 1	Byte 0	Primary Offset	Secondary Offset
Downstream Configuration Address[1]				80h	80h
Downstream Configuration Data[1]				84h	84h
Upstream Configuration Address[1]				88h	88h
Upstream Configuration Data[1]				8Ch	8Ch
Configuration Control and Status[1]		Configuration Own Bits[1]		90h	90h
Downstream Memory 0 Translated Base[1]				94h	94h
Downstream I/O or Memory 1 Translated Base[1]				98h	98h
Downstream Memory 2 Translated Base[1]				9Ch	9Ch
Downstream Memory 3 Translated Base[1]				A0h	A0h
Upstream I/O or Memory 0 Translated Base[1]				A4h	A4h
Upstream Memory 1 Translated Base[1]				A8h	A8h
Downstream Memory 0 Setup Register[2]				ACh	ACh
Downstream I/O or Memory 1 Setup Register[2]				B0h	B0h
Downstream Memory 2 Setup Register[2]				B4h	B4h
Downstream Memory 3 Setup Register[2]				B8h	B8h
Upper 32 Bits Downstream Memory 3 Setup Register[2]				BCh	BCh
Primary Expansion ROM Setup Register[2]				C0h	C0h
Upstream I/O or Memory 0 Setup Register[2]				C4h	C4h
Upstream Memory 1 Setup Register[2]				C8h	C8h
Chip Control 1[2]		Chip Control 0[2]		CCh	CCh
Arbiter Control[2]		Chip Status		D0h	D0h
Reserved		Secondary SERR# Disables[2]	Primary SERR# Disables[2]	D4h	D4h
Reset Control				D8h	D8h
Power Management Capabilities[2]		Next Item Ptr	Capability ID	DCh	DCh
PM Data[2]	PMCSR BSE	Power Management CSR[2]		E0h	E0h
VPD Address		Next Item Ptr	Capability ID	E4h	E4h
VPD Data				E8h	E8h
Reserved	Hot Swap Control	Next Item Ptr	Capability ID[2]	ECh	ECh
Reserved				FF:F0h	FF:F0h

1. Shared mapping in the 21554 I/O or memory space.

2. Register or a portion of the register may be preloaded using the serial ROM interface.

FM-06136.AI4

7.2 Configuration Register Description

The following subsections describe the 21554 configuration registers.

- Primary and secondary configuration header registers, describing both the shared and separate standard PCI configuration registers, with the exception of the base address registers
- Primary and secondary addressing registers, including base address registers, translation registers, and setup registers
- Registers for upstream and downstream configuration transaction generation
- Device-specific control and status registers

If a register is associated with the primary interface, then its name is preceded with *Primary.* If a register is associated with the secondary interface, then its name is preceded with *Secondary.* If a register is shared by both interfaces, then it is not preceded with *Primary* or *Secondary.* The byte offsets at which each register can be accessed from each interface are listed in each register description.

Table 7-1 shows the allowable access to each register.

Table 7-1. Register Access

Abbreviation	Definition
R	Read only. Writes have no effect.
R/W	Read/write
R/W1TC	Read. Write 1 to clear.
R/W1TS	Read. Write 1 to set.
R0TS	Read 0 to set.
R/(WS)	Read. Write from secondary interface only. Primary bus writes have no effect.
R/(WP)	Read. Write from primary interface only. Secondary bus writes have no effect.

7.2.1 Shared Standard PCI Registers

The registers described in this section are shared between the primary and secondary interfaces.

7.2.1.1 Vendor ID Register

Primary byte offset: 01:00h and 41:40h

Secondary byte offset: 41:40h and 01:00h

Bit	Name	R/W	Description
15:0	VendorID	R	The Vendor ID identifies Intel as the vendor of this device and is internally hardwired to be 1011 hex.

7.2.1.2 Device ID Register

Primary byte offset: 03:02h and 43:42h

Secondary byte offset: 43:42h and 03:02h

Bit	Name	R/W	Description
15:0	DeviceID	R	Device ID identifies this device as the 21554 and is internally hardwired to be 46h.

7.2.1.3 Revision ID (RevID) Register

Primary byte offset: 08h and 48h

Secondary byte offset: 08h and 48h

Bit	Name	R/W	Description
7:0	RevisionID	R	When read, the Revision ID indicates the revision number of this device. The initial revision reads as 0. Subsequent revisions increment by 1.

7.2.1.4 BiST Register

Primary byte offset: 0Fh and 4Fh

Secondary byte offset: 0Fh and 4Fh

The 21554 does not implement self-test internally and does not directly use the BiST register. However, some form of self-test may be desired in the subsystem so mechanisms are provided by the 21554 to support vendor-specific usage of the BiST register. The default value of this register is 00h after reset assertion, which indicates that BiST is not supported. However, after reset the 21554 allows this field to be automatically preloaded with a value from the serial ROM (if attached) or programmed via the secondary interface by the local processor.

Bit	Name	R/W	Description
3:0	Completion Code	R/(WS)	The completion code can only be written by the secondary interface (at secondary offset 0Fh or offset 4Fh). A Completion Code value of 0h indicates that the device passed its self-test. Any non-zero value in the Completion Code indicates that the device failed its self-test.
5:4	Reserved	R	Reserved. Read only as 0.
6	Self Test	R/W	This bit can be written via the primary interface or secondary interface configuration registers. Configuration code running on the host processor sets this bit to 1 to invoke self-test. The local processor (or some other device) on the secondary interface clears this bit to 0 to indicate the completion of the self-test (after first updating the Completion Code bit field).
7	BiST Supported	R/(WS)	This bit can be written by the secondary interface (at secondary offset 0Fh or offset 4Fh) or it may be preloaded using the serial ROM. A value of 1 indicates to configuration software that the device supports self-test. A value of 0 indicates that the device does not support self-test.

7.2.1.5 Header Type Register

Primary byte offset: 0Eh and 4Eh

Secondary byte offset: 0Eh and 4Eh

Bit	Name	R/W	Description
7:0	Header Type	R	Defines the layout of addresses 00h through 3Fh in configuration space. Reads as 00h indicating a Type 0 header format.

7.2.1.6 Subsystem Vendor ID Register

Primary byte offset: 2D:2Ch and 6D:6Ch

Secondary byte offset: 6D:6Ch and 2D:2Ch

Bit	Name	R/W	Description
15:0	Subsystem Vendor ID	R/(WS)	Identifies the vendor of the add-in card or subsystem. This register is initialized by either the local processor or by serial ROM preload.

7.2.1.7 Subsystem ID Register

Primary byte offset: 2F:2Eh and 6F:6Eh

Secondary byte offset: 6F:6Eh and 2F:2Eh

Bit	Name	R/W	Description
15:0	Subsystem ID	R/(WS)	Identifies the vendor-specific device ID for subsystem. This register is initialized by either the local processor or by serial ROM preload.

7.2.2 Primary and Secondary Standard PCI Registers

The register types in this section have separate registers for the primary and secondary interfaces. However, the register description is given once, and applies to both the primary and secondary configuration registers. The primary register controls behavior on the primary interface only, and the secondary register controls behavior on the secondary interface only.

7.2.2.1 Primary and Secondary Command Registers

	Primary Command	Secondary Command
Primary byte offset:	05:04h	45:44h
Secondary byte offset:	45:44h	05:04h

Table 7-2. Primary and Secondary Command Registers (Sheet 1 of 2)

Bit	Name	R/W	Description
0	I/O Space Enable	R/W	Controls 21554 response to I/O transactions on the corresponding interface. *When 0:* The 21554 does not respond to I/O transactions. *When 1*: The 21554 response to I/O transactions is enabled. *Reset value*: 0
1	Memory Space Enable	R/W	Controls response to memory transactions on the corresponding interface. *When 0:* The 21554 does not respond to memory transactions. *When 1:* The 21554 response to memory transactions is enabled. *Reset value:* 0.
2	Master Enable	R/W	Controls 21554's ability to initiate memory and I/O transactions on the corresponding interface. Initiation of configuration transactions is not affected. *When 0:* The 21554 does not initiate memory or I/O transactions. *When 1:* The 21554 is enabled to operate as an initiator. *Reset value*: 0.
3	Special Cycle Enable	R	The 21554 ignores special cycle transactions, so this bit is read only and returns 0.
4	Memory Write and Invalidate Enable	R/W	This bit controls the ability of the 21554 to generate Memory Write and Invalidate (MWI) bus commands as a master on the corresponding interface. *When 0:* Disables use of Memory Write and Invalidate bus commands (uses Memory Write commands instead). *When 1:* Enables use of Memory Write and Invalidate bus commands. *Reset Value:* 0
5	VGA Snoop Enable	R	Reads only as 0 to indicate the 21554 does not respond to VGA palette writes.
6	Parity Error Response	R/W	Controls the response of the 21554 when a parity error is detected on the corresponding interface. *When 0:* The 21554 does not assert PERR#, nor does it set the Data Parity Reported bit in the appropriate Primary or Secondary Status registers. The 21554 does not report address parity errors by asserting SERR#. *When 1:* The 21554 drives PERR# and conditionally sets the Data Parity Reported bit in the Primary or Secondary Status register when a data parity error is detected. The 21554 allows SERR# assertion when address parity errors are detected. *Reset value:* 0.

Table 7-2. Primary and Secondary Command Registers (Sheet 2 of 2)

Bit	Name	R/W	Description
7	Wait Cycle Control	R	Reads as zero to indicate the 21554 does not perform address or data stepping.
8	SERR# Enable	R/W	Controls the enable for SERR# on the corresponding interface. *When 0:* SERR# cannot be driven by the 21554. *When 1:* SERR# may be driven low by the 21554 under the conditions described in Chapter 14. *Reset value:* 0.
9	Fast Back-to-Back Enable	R/W	Controls the ability of the 21554 to generate fast back-to-back transactions on the corresponding bus. *When 0:* The 21554 does not generate back-to-back transactions. *When 1:* The 21554 is enabled to generate back-to-back transactions. *Reset value:* 0.
15:10	Reserved	R	Reserved. Returns 0 when read.

7.2.2.2 Primary and Secondary Status Registers

	Primary Status	Secondary Status
Primary byte offset:	07:06h	47:46h
Secondary byte offset:	47:46h	07:06h

The bits described below reflect the status of the 21554 primary interface for the Primary Status register, and of the secondary interface for the Secondary Status register. W1TC indicates that writing a 1 to that bit clears the bit to 0. Writing a 0 has no effect.

Table 7-3. Primary and Secondary Status Registers

Bit	Name	R/W	Description
3:0	Reserved	R	Reserved. Returns 0 when read.
4	ECP Support	R	Enhanced Capabilities Support Indicator. Reads as 1 to indicate that the Enhanced Capabilities Port is supported.
5	66 MHz Capable	R	66 MHz Capable Indication. Reads as 0 to indicate that the corresponding interface operates at a maximum frequency of 33 MHz.
6	Reserved	R	Reserved. Returns 0 when read.
7	Fast Back-to-Back Capable	R	Reads as 1 to indicate that the 21554 is able to respond to fast back-to-back transactions on the corresponding interface.
8	Data Parity Detected	R/ W1TC	This bit is set to a 1 when all of the following are true: • The 21554 is a master on the corresponding bus. • PERR# is detected asserted for writes or a parity error is detected for reads. • Parity Error Response bit is set in the Primary or Secondary Command register. *Reset value:* 0.
10:9	DEVSEL# timing	R	Indicates slowest response to a nonconfiguration command on the corresponding interface. Reads as 01b to indicate that the 21554 responds no slower than with medium timing.
11	Signaled Target Abort	R/ W1TC	This bit is set to a 1 when the 21554 is acting as a target on the corresponding bus and returns a target abort to the initiator. *Reset value:* 0.
12	Received Target Abort	R/ W1TC	This bit is set to a 1 when the 21554 is acting as an initiator on the corresponding bus and receives a target abort. *Reset value:* 0.
13	Received Master Abort	R/ W1TC	This bit is set to a 1 when the 21554 is acting as an initiator on the corresponding bus and detects a master abort. *Reset value:* 0.
14	Signaled System Error	R/ W1TC	This bit is set to a 1 when the 21554 has asserted SERR# on the corresponding bus. *Reset value:* 0.
15	Detected Parity Error	R/ W1TC	This bit is set to a 1 when the 21554 detects an address or data parity error on the corresponding interface. *Reset value:* 0.

7.2.2.3 Primary and Secondary Class Code Registers

	Primary Class Code	Secondary Class Code
Primary byte offset:	0B:09h	4B:49h
Secondary byte offset:	4B:49h	0B:09h

These registers may be preloaded through the serial ROM. The Primary Class Code register may also be programmed by the local processor.

Table 7-4. Primary and Secondary Class Code Registers

Bit	Name	R/W	Description
7:0	ProgIF (PIF)	PPIF: R/(WS) SPIF: R	Reads as zero.
15:8	Sub-Class Code (SCC)	PSCC: R/(WS) SSCC: R	Reads as 80 hex to indicate that this bridge device is classified as “other”.
23:16	Base Class Code (BCC)	PBCC:R/(WS) SBCC: R	Reads as 06 hex to indicate device is a bridge device.

7.2.2.4 Primary and Secondary Cache Line Size Registers

	Primary Cache Line Size	Secondary Cache Line Size
Primary byte offset:	0Ch	4Ch
Secondary byte offset	4Ch	0Ch

Table 7-5. Primary and Secondary Cache Line Size Registers

Bit	Name	R/W	Description
7:0	Cache Line Size	R/W	Designates the cache line size for the corresponding interface in units of 32-bit Dwords. Used for prefetching memory reads and for terminating memory write and invalidates. Valid cache line sizes are 4, 8, 16, and 32 Dwords. If the cache line size is set to any other value, the 21554 uses the same behavior as when the cache line size is set to 8. *Reset value:* 00h.

7.2.2.5 Primary Latency and Secondary Master Latency Timer Registers

	Primary MLT	Secondary MLT
Primary byte offset:	0Dh	4Dh
Secondary byte offset:	4Dh	0Dh

Table 7-6. Primary Latency and Secondary Master Latency Timer Registers

Bit	Name	R/W	Description
7:0	Master Latency Timer	R/W	Master latency timer for the corresponding interface. Indicates the number of PCI clock cycles from the assertion of FRAME# to the expiration of the timer when the 21554 is acting as a master. All bits are writeable, resulting in a granularity of 1 PCI clock cycle. *When 0:* The 21554 relinquishes the bus after the first data transfer when the 21554’s PCI bus grant has been deasserted. *Reset value:* 00h.

7.2.2.6 Primary and Secondary Interrupt Line Registers

	Primary Interrupt Line	Secondary Interrupt Line
Primary byte offset:	3Ch	7Ch
Secondary byte offset:	7Ch	3Ch

Table 7-7. Primary and Secondary Interrupt Line Registers

Bit	Name	R/W	Description
7:0	Interrupt Line	R/W	This register is used to communicate interrupt line routing information for the corresponding interface. This register must be initialized by initialization code so a default state after reset assertion is not specified. Initialization code writes this register with a value indicating to which input of the system interrupt controller the 21554 bus interrupt signal pin INTA# is connected.

7.2.2.7 Primary and Secondary Interrupt Pin Registers

	Primary Interrupt Pin	Secondary Interrupt Pin
Primary byte offset:	3Dh	7Dh
Secondary byte offset:	7Dh	3Dh

Table 7-8. Primary and Secondary Interrupt Pin Registers

Bit	Name	R/W	Description
7:0	Interrupt Pin	R	This register indicates which PCI interrupt pin the 21554 uses on the corresponding bus. This is a read-only register and always returns 1 when read indicating that the 21554 uses INTA#.

7.2.2.8 Primary and Secondary Minimum Grant Registers

	Primary Minimum Grant	Secondary Minimum Grant
Primary byte offset:	3Eh	7Eh
Secondary byte offset:	7Eh	3Eh

These registers may be preloaded through the serial ROM. The Primary Minimum Grant register may also be programmed by the local processor.

Table 7-9. Primary and Secondary Minimum Grant Registers

Bit	Name	R/W	Description
7:0	MIN_GNT (MG)	PMG: R/(WS) SMG: R	Specifies how long of a burst period the 21554 needs on the corresponding bus in units of 1/4 µsec. Reads as 0 before preload.

7.2.2.9 Primary and Secondary Maximum Latency Registers

	Primary Maximum Latency	Secondary Maximum Latency
Primary byte offset:	3Fh	7Fh
Secondary byte offset:	7Fh	3Fh

These registers may be preloaded through the serial ROM. The Primary Maximum Latency register may also be programmed by the local processor.

Table 7-10. Primary and Secondary Maximum Latency Registers

Bit	Name	R/W	Description
7:0	MAX_LAT (ML)	PML: R/(WS) SML: R	Specifies how often the 21554 needs to gain access to the corresponding bus in units of 1/4 µsec. Reads as 0 before preload.

7.2.2.10 Enhanced Capabilities Pointer Register

	ECP
Primary byte offset:	34h and 74h
Secondary byte offset:	34h and 74h

Table 7-11. Enhanced Capabilities Pointer Register

Bit	Name	R/W	Description
7:0	ECP	R	Pointer to the first set of ECP registers. Returns DCh to indicate that the first set of ECP registers begins at configuration offset DCh. For the 21554, this points to the Power Management registers. *Reset value:* DCh

7.2.2.11 Power Management Capability ID Register

Primary byte offset: DCh
Secondary byte offset: DCh

Table 7-12. Power Management Capability ID Register

Bit	Name	R/W	Description
7:0	PM ECP ID	R	Power Management Enhanced Capabilities Port ID. Read only as 01h to identify these ECP registers as Power Management registers.

7.2.2.12 Power Management Next Item Pointer Register

Primary byte offset: DDh
Secondary byte offset: DDh

Table 7-13. Power Management Next Item Pointer Register

Bit	Name	R/W	Description
7:0	PM Next Ptr	R	Pointer to next ECP registers. Reads as E4 to indicate the first register of the next set of ECP registers, which support Vital Product Data, is located at offset E4h.

7.2.2.13 Power Management Capabilities Register

Primary byte offset: DF:DEh
Secondary byte offset: DF:DEh

Bits [14:9,5,2:0] are loadable through the serial ROM or are programmable by the local processor.

Table 7-14. Power Management Capabilities Register

Bit	Name	R/W	Description
2:0	PM Version	R/(WS)	Power Management Version. Loadable by serial ROM. *Reset value:* 001b to indicate that this device is compliant to Revision 1.0 of the PCI Power Management Interface Specification.
3	PME Clock	R	Clock Required for PME# Assertion. Reads as 1 to indicate that a clock is required to assert PME#, if any of bits [15:11] in this register is asserted. Read as 0 if bits [15:11] are all 0, indicating that the 21554 does not assert PME#.
4	APS	R	Auxiliary Power Source. Not defined since the 21554 does not have PME# support from $D3_{cold}$. Read only as 0.
5	DSI	R/(WS)	Device-Specific Initialization. Loadable by serial ROM. *When 0:* Indicates that the 21554 does not have device-specific initialization requirements. *When 1:* Indicates that the 21554 has device-specific initialization requirements. *Reset value:* 0
8:6	Reserved	R	Reserved. Read only as 0.
9	D1 Support	R/(WS)	D1 Power State Support Indicator. Loadable by serial ROM. *When 0*: Indicates that the 21554 does not support the D1 power management state. *When 1:* Indicates that the 21554 supports the D1 power management state. *Reset value:* 0
10	D2 Support	R/(WS)	D2 Power State Support Indicator. Loadable by serial ROM. *When 0*: Indicates that the 21554 does not support the D2 power management state. *When 1:* Indicates that the 21554 supports the D2 power management state. *Reset value:* 0
15:11	PME Support	R/(WS)	PME# support. Indicates whether the 21554 asserts **p_pme_l** when in a given power state. Bit 11 corresponds to D0; bit 15 corresponds to $D3_{cold}$. Bits [14:11] are loadable by serial ROM. Bit 15 always reads as 0 and is not loaded by serial ROM nor writeable from the secondary interface, since the 21554 never asserts PME# when in $D3_{cold}$. *Reset value:* 0 to indicate that the 21554 does not implement the PME# pin.

7.2.2.14 Power Management Control and Status Register

Primary byte offset: E1:E0h
Secondary byte offset: E1:E0h

Bits [14:13] are loadable by serial ROM or are programmable by the local processor.

Table 7-15. Power Management Control and Status Register

Bit	Name	R/W	Description
1:0	PWR State	R/W	Power State. Reflects the current power state of the 21554. If an unimplemented power state is written to this register, the 21554 completes the write transaction, ignores the write data, and does not change the value of this field. D0 and D3 are always implemented. Support of D1 and D2 is determined by serial ROM preload or local processor. 00b: D0 (required) 01b: D1 (optional) 10b: D2 (optional) 11b: D3 (required) *Reset value:* 00b
3:2	Reserved	R	Reserved. Read only as 00b.
4	DYN DATA	R	Dynamic Data. Reads as 0 to indicate that the 21554 does not support dynamic data reporting.
7:5	Reserved	R	Reserved. Read only as 000b.
8	PME_EN	R/W	PME# Enable. Read only as 0 if the PME Support bits are all 0. Otherwise, this is a read/write bit. *When 0:* The 21554 will not assert p_pme_l. *When 1:* The 21554 is enabled to assert p_pme_l. *Reset value:* 0
12:9	DATA_SEL	R/W	Data Select. This register is enabled by loading a “1” for the PM Data Enable function in the serial ROM. For values of 7:0, selects one of eight bytes of data loaded by serial ROM to be placed in the power management data register. For values of 15:8, a 0 is returned in the data register. If not enabled, this register always returns 0 when read. *Reset value:* 000b.
14:13	Data Scale	R/(WS)	Data Scale. Indicates the scaling factor of the value in the power management data register. Loadable by serial ROM. *Reset value:* 00b
15	PME Status	R/W1TC	PME Status. The 21554 sets this bit to a 1 when s_pme_l is asserted and the PME# Support bit for the current power state is a 1. This corresponds to when the 21554 would normally assert p_pme_l, but regardless of the state of the PME_En bit. Writing a 1 clears this bit. Writing a 0 has no effect. *Reset value:* 0.

7.2.2.15 PMCSR Bridge Support Extensions

Primary byte offset: E2h
Secondary byte offset: E2h

Table 7-16. PMCSR Bridge Support Extensions

Bit	Name	R/W	Description
7:0	BSE	R	Bridge Support Extensions. Read only as 00h.

7.2.2.16 Power Management Data Register

Primary byte offset: E3h
Secondary byte offset: E3h

Table 7-17. Power Management Data Register

Bit	Name	R/W	Description
7:0	PM Data	R	Power Management Data register. Reflects one of eight bytes loaded by serial ROM, or reads as 0. Bytes are selected by the data select register. *Reset value:* 00h

7.2.2.17 Vital Product Data (VPD) ECP Register

Primary byte offset: E4h
Secondary byte offset: E4h

Table 7-18. Vital Product Data (VPD) ECP Register

Bit	Name	R/W	Description
7:0	50	R	VPD Enhanced Capabilities Port ID. Read only as 03h to identify these ECP registers as VPD registers.

7.2.2.18 Vital Product Data (VPD) Next Pointer Register

Primary byte offset: E5h
Secondary byte offset: E5h

Table 7-19. Vital Product Data (VPD) Next Pointer Register

Bit	Name	R/W	Description
7:0	VPD Next Ptr	R	Pointer to next ECP registers. Reads as ECh to point to the next set of ECP registers, supporting Compact PCI Hot Swap.

7.2.2.19 Vital Product Data (VPD) Address Register

Primary byte offset: E7:E6h
Secondary byte offset: E7:E6h

Table 7-20. Vital Product Data (VPD) Address Register

Bit	Name	R/W	Description
8:0	VPD Addr	R/W	Vital Product Data Address. Contains the VPD byte address of the serial ROM location to be accessed. Valid VPD byte addresses are 17F: 000h. VPD starts at base address 080h in the serial ROM. The VPD byte address contained in this register is added to the VPD base address to obtain the final serial ROM address.
14:9	Reserved	R	Reserved. Read Only as 0.
15	VPD Flag	R/W	VPD Flag. Starts a VPD serial ROM access and indicates completion of the operation. *When written with a 0:* A 4-byte serial ROM read is performed starting at the VPD location indicated by bits [8:0]. Note that this operation is not necessarily Dword aligned. When the read is complete, the 21554 sets this bit to a 1. *When written with a 1:* A 4-byte serial ROM write is performed starting at the VPD location indicated by bits [8:0]. Note that this operation is not necessarily Dword aligned. When the write is complete, the 21554 sets this bit to a 0.

7.2.2.20 VPD Data Register

Primary byte offset: EB:E8h
Secondary byte offset: EB:E8h

Table 7-21. VPD Data Register

Bit	Name	R/W	Description
31:0	VPD Data	R/W	VPD Data. Contains the VPD read or write data. For a read, this register should be read after a read operation was initiated and the 21554 has returned the VPD Flag bit to a 1. For a write, this register should be written with the write data before the operation is initiated with a write to the VPD Address and VPD Flag bits. VPD read and write operations are always 4-byte operations. Byte 0 contains the data corresponding to the starting VPD byte address. Byte 1, 2, and 3 contain successive bytes. Note that Byte 0 is not necessarily Dword aligned.

7.2.2.21 Compact PCI Hot-Swap Capability Identifier Register

Primary byte offset: ECh
Secondary byte offset: ECh

Table 7-22. Compact PCI Hot-Swap Capability Identifier Register

Bit	Name	R/W	Description
7:0	HS ECP ID	R	Enhanced capabilities ID. Reads only as 06h to indicate that these are Compact PCI Hot-Swap registers.

7.2.2.22 Compact PCI Hot-Swap Next Pointer Register

Primary byte offset: EDh
Secondary byte offset: EDh

Table 7-23. Compact PCI Hot-Swap Next Pointer Register

Bit	Name	R/W	Description
7:0	HS NXT PTR	R	Pointer to next set of ECP registers. Reads only as 0 to indicate that these are the last ECP registers in this list.

7.2.2.23 Compact PCI Hot-Swap Control Register

Primary byte offset: EF:EEh
Secondary byte offset: EF:EEh

Table 7-24. Compact PCI Hot-Swap Control Register

Bit	Name	R/W	Description
0	Reserved	R	Reserved. Read only as 0.
1	ENUM_MASK	R/W	ENUM# Interrupt Mask. *When 0:* The 21554 asserts p_enum_l when an insertion or removal event occurs. *When 1:* The 21554 does not assert p_enum_l. Reset value: 0
2	Reserved	R	Reserved. Read only as 0.
3	LED On/Off (LOO)	R/W	LED On/Off (LOO) Control. Allows software control of the l_stat pin and therefore the state of the LED. *When 0:* The 21554 tristates l_stat. If REM STAT is low, the LED is off if the ejector handle is closed and on if the ejector handle is open. If REM STAT is high, l_stat is not tristated but continues to be driven by the 21554 (LED is off). *When 1:* The 21554 drives l_stat high and the LED is forced on. *Reset value:* 0
5:4	Reserved	R	Returns 0 when read.
6	REM STAT	R/W1 TC	Signal p_enum_l Removal Status. The 21554 sets this bit to a 1 when l_stat is sampled high and p_rst_l is deasserted, signaling an impending removal. This bit is cleared when software writes a 1. Clearing this bit causes the 21554 to tri-state l_stat. Writing a 0 has no effect. *When 1:* The 21554 is asserting p_enum_l to indicate that the card is about to be removed. Reset value: 0
7	INS STAT	R/W1 TC	Signal p_enum_l Insertion Status. The 21554 sets this bit to a 1 when l_stat is sampled low (ejector handle is closed), the serial preload is complete and the Primary Lockout bit is cleared, indicating that the card is ready for host initialization. This bit is cleared when software writes a 1. Writing a 0 has no effect. *When 1:* The 21554 is asserting p_enum_l to indicate that the card has just been inserted. Reset value: 0

7.2.3 Primary and Secondary Address Registers

This section contains descriptions of primary and secondary address registers.

7.2.3.1 Primary CSR and Downstream Memory 0 Base Address Register

Primary byte offset: 13:10h
Secondary byte offset: 53:50h

The Primary CSR and Downstream Memory 0 Base Address registers are used to map the 21554 registers into primary memory space, and optionally to specify a memory range for downstream forwarding of memory transactions. In order to specify a downstream forwarding range, the Downstream Memory 0 Setup Register must be loaded either from the serial ROM or by the local processor before configuration software running on the host processor can access this register. Local processor access of the setup register should be done before the Primary Lockout flag is cleared.

Table 7-25. Primary CSR and Downstream Memory 0 Base Address Register

Bit	Name	R/W	Description
0	Space Indicator	R	Type of address space requested. Reads as a 0 to indicate that memory space is requested.
2:1	Type	R	Indicates size and location of this address space. *Reset value:* To 00 to indicate that this space can be mapped anywhere in 32-bit memory.
3	Prefetchable	R	Indicates whether the region is prefetchable. Accesses to the 21554 registers are disconnected after the first data phase. *When 0:* Nonprefetchable memory is requested. *When 1:* Prefetchable memory is requested. Reset value: 0
11:4		R	Returns 0 when read.
31:12	Base Address	R/W	These bits are used to indicate the size of the requested address range and to set the base address of the range. The low 4 KB of this address range map the 21554 CSRs into primary memory space. The remaining space in this range above 4 KB, if any, specifies a range for downstream forwarding of memory transactions. Bits [30:12] of the Downstream Memory 0 Setup register determine the function of the corresponding bit in this register. If a bit in the setup register is 0 then the same bit in this register is a read-only bit and always returns 0 when read. If a bit in the setup register is 1 then the same bit in this register is writeable and returns the value last written when read. If the setup register is written to all zeros, the minimum size of 4 KB is requested (the 21554 CSR access only, no forwarding range). The maximum size of this range is 2 GB; therefore bit [31] is always writeable. *Reset value:* 4 KB of nonprefetchable memory requested.

7.2.3.2 Secondary CSR Memory Base Address Registers

Primary byte offset: 53:50h
Secondary byte offset: 13:10h

The Secondary CSR Memory Base Address register is used to map the 21554 registers into secondary bus memory space.

Table 7-26. Secondary CSR Memory Base Address Registers

Bit	Name	R/W	Description
0	Space Indicator	R	Type of address space requested. Reads as a 0 to indicate that memory space is requested.
2:1	Type	R	Indicates size and location of the 21554 memory mapped registers. Reads as 00 to indicate that the 21554 registers can be mapped anywhere in 32-bit memory address space.
3	Prefetchable	R	Indicates whether this space is prefetchable. Reads as 0 to indicate that prefetching should not be used when reading the 21554 registers.
11:4		R	Returns 0 when read.
31:12	Base Address	R/W	These bits are used to communicate to configuration software the size of the requested memory address range and to set the base address of the range. Since bits [31:12] are mappable, this indicates that the 21554 is requesting 4KB of memory space.

7.2.3.3 Primary and Secondary CSR I/O Base Address Register

	Primary CSR I/O BAR	Secondary CSR I/O BAR
Primary byte offset:	17:14h	57:54h
Secondary byte offset:	57:54h	17:14h

The Primary and Secondary CSR I/O Base Address registers are used to map the 21554 registers into primary and secondary I/O space, respectively.

Table 7-27. Primary and Secondary CSR I/O Base Address Register

Bit	Name	R/W	Description
0	Space Indicator	R	Type of address space requested. Reads as a 1 to indicate that I/O space is requested.
7:1	Reserved	R	Reserved. Returns 0 when read.
31:8	Base Address	R/W	These bits are used to communicate to configuration software the size of the requested I/O address range and to set the base address of the range. Since bits [31:8] are mappable, this indicates that the 21554 is requesting 256 bytes of I/O space.

7.2.3.4 Downstream I/O or Memory 1 and Upstream I/O or Memory 0 BAR

	Downstream I/O or Memory 1 BAR	Upstream I/O or Memory 0 BAR
Primary byte offset:	1B:18h	5B:58h
Secondary byte offset:	5B:58h	1B:18h

These registers define address ranges in which either I/O or memory transactions can be forwarded downstream or upstream. After reset, these registers are disabled and return all zeros when read. This register may be enabled to request either 64, 128, or 256 bytes of I/O space (although hardware does not restrict larger I/O windows), or 4 KB to 2 GB of memory space. The amount of space requested is configured either by serial preload or by programming the Downstream I/O or Memory 1 Setup register (for the downstream BAR) or the Upstream I/O or Memory 0 Setup register (for the upstream BAR). Local processor access of the setup registers should be done before the Primary Lockout flag is cleared.

Table 7-28. Downstream I/O or Memory 1 and Upstream I/O or Memory 0 BAR

Bit	Name	R/W	Description
0	Space Indicator	R	*When read as a 0*: Indicates that this BAR is disabled or is requesting memory space. *When read as a 1:* Indicates that I/O space is requested. *Reset value:* 0
2:1	Type	R	Type indicator. If requesting I/O space, these bits read as zeros. If requesting memory space, these bits indicate size and location of this address range. *Reset value:* 00b
3	Prefetchable	R	Prefetchable indicator. If requesting I/O or nonprefetchable memory, reads as 0. If requesting prefetchable memory space, reads as 1. *Reset value:* 0
5:4		R	Read only. Returns 0 when read.
31:6	Base Address	R/W	These bits are used to indicate the size of the requested address range and to set the base address of the range. Bits [30:6] of the corresponding setup register determine the function of the corresponding bit in this register. If a bit in the setup register is 0 then the same bit in this register is a read-only bit and always return 0 when read. If a bit in the setup register is 1, then the same bit in this register is writeable and return the value last written when read. This base address register may be disabled by writing bit [31] of the setup register to zero. The minimum size for an I/O address range is 64 bytes and for a memory range is 4 KB. The maximum size is 2 GB. *Reset value:* This register is disabled (read only as 0).

7.2.3.5 Downstream Memory 2 and 3 BAR, Upstream Memory 1 BAR

Because the description for these registers is very similar, they are discussed together.

	Downstream Memory 2 BAR	Downstream Memory 3 BAR	Upstream Memory 1 BAR
Primary byte offset:	1F:1Ch	23:20h	5F:5Ch
Secondary byte offset:	5F:5Ch	63:60h	1F:1Ch

These registers define address ranges in which memory transactions on the primary interface of the 21554 are forwarded to the secondary interface for the downstream BARs, and in which memory transactions on the secondary interface are forwarded to the primary interface for the upstream BAR. The setup registers corresponding to these BARs must be loaded either from the serial ROM or by the local processor before configuration software running on the host processor can access these registers. Local processor access of the setup registers should be done before the Primary Lockout flag is cleared.

Table 7-29. Downstream Memory 2 and 3 BAR, Upstream Memory 1 BAR

Bit	Name	R/W	Description
0	Space Indicator	R	Reads only as 0 to indicate that memory space is requested.
2:1	Type	R	Indicates size and location of this address space. *Reset value:* 00 to indicate that this space can be mapped anywhere in 32-bit memory.
3	Prefetchable	R	Indicates whether the region is prefetchable. *When 0:* Nonprefetchable memory is requested (or the range is disabled). *When 1:* Prefetchable memory is requested. Reset value: 0
11:4		R	Read only. Returns 0 when read.
31:12	Base Address	R/W	These bits are used to indicate the size of the requested address range and to set the base address of the range. Bits [31:12] of the corresponding setup register determine the function of the corresponding bit in this register. If a bit in the setup register is 0, then the same bit in this register is a read only bit and always returns 0 when read. If a bit in the setup register is 1, then the same bit in this register is writeable and returns the value last written when read. This base address register is disabled by writing bit [31] of the setup register to zero. For the Downstream Memory 3 BAR, bit [31] of the Upper 32 Bits setup register must also be 0 to disable that range. The minimum size for this address range is 4 K. The maximum size is 2GB, except for the Downstream Memory 3 BAR which may use 64-bit addressing and have a maximum window size of 2^{63} bytes. *Reset value:* Read only as 0 (range is disabled).

7.2.3.6 Upper 32 Bits Downstream Memory 3 Base Address Register

Primary byte offset: 27:24h
Secondary byte offset: 67:64h

Table 7-30. Upper 32 Bits Downstream Memory 3 Base Address Register

Bit	Name	R/W	Description
31:0	Base Address	R/W	This register defines the upper 32 bits of a memory range for downstream forwarding of memory transactions. The lower 32 bits are contained in the Downstream Memory 3 Base Address register. These bits are used to indicate the size of the requested address range and to set the base address of the range. The value of each bit in the Upper 32 Bits Downstream Memory 3 Setup register determines the function of the corresponding bit in this register. If a bit in the setup register is 0, then the same bit in this register is a read-only bit and always return 0 when read. If a bit in the setup register is 1, then the same bit in this register is writeable and returns the value last written when read. This base address register is disabled when bit [31] of the Upper 32 Bits Downstream Memory 3 Base Address register is 0. The minimum size for this address range is 4 K. The maximum size is 2^{63} bytes. *Reset value:* Read only as 0 (range is disabled).

7.2.3.7 Upstream Memory 2 Base Address Register

Primary byte offset: 63:60h
Secondary byte offset: 23:20h

This register defines the memory range for upstream forwarding of transactions using lookup table based address translation. The size of this register is programmed or disabled by setting the page size in the Chip Control 1 configuration register.

Table 7-31. Upstream Memory 2 Base Address Register

Bit	Name	R/W	Description
0	Space Indicator	R	Reads only as 0 to indicate that memory space is requested.
2:1	Type	R	Indicates size and location of this address space. Reads as 00 to indicate that this space can be mapped anywhere in 32-bit memory.
3	Prefetchable	R	When this address range is enabled, read only as 1h to indicate prefetchable memory. Page entries also may be individually designated as prefetchable or nonprefetchable, where a nonprefetchable entry overrides this prefetchable bit.
13:4		R	Read Only. Returns 0 when read.
31:14	Base Address	R/W	These bits are used to indicate the size of the requested address range and to set the base address of the memory range for upstream forwarding using lookup table based address translation. The size of this window is a function of the page size, and can vary from 16 KB to 4 MB increasing by powers of two. The number of writeable bits is dependent on the window size, and varies from [31:14] for a 16 KB window to [31:28] for a 256 MB window. *Reset value:* This address range is disabled (reads only as 0).

7.2.3.8 Primary Expansion ROM Base Address Register

Primary byte offset: 33:30h
Secondary byte offset: 73:70h

This register defines an address range in which a memory read transaction on the primary interface of the 21554 results in a read access to the parallel ROM interface. The Primary Expansion ROM Setup register controls the size of the address range requested by the Primary Expansion ROM Base Address register. The Primary Expansion ROM Setup register must be loaded either from the serial ROM or by the local processor before configuration software running on the host processor can access this register. Local processor access should be done before the configuration lockout flag is cleared.

Table 7-32. Primary Expansion ROM Base Address Register

Bit	Name	R/W	Description
0	Address Decode Enable	R/W	Enables the 21554 to respond to accesses to its expansion ROM space. If this BAR is disabled this bit will return zero when read. *When 1:* The 21554 responds to memory accesses to expansion ROM space if the Memory Enable bit is also set. *When 0:* The 21554 does not respond to accesses directed to this address space. *Reset value:* 0.
11:1	Reserved	R	Reserved. Returns 0 when read.
31:12	Base Address	R/W	These bits are used to indicate the size of the expansion ROM space and to set the base address of the range. Bits [23:11] of the Primary Expansion ROM Setup register determine the function of the corresponding bit in this register. If a bit in the Primary Expansion ROM Setup register is 0, the same bit in this register is a read-only bit and always returns 0 when read. If a bit in the Primary Expansion ROM Setup register is 1, then the same bit in this register is writeable and returns the value last written when read. When this BAR is enabled, bits [31:24] are always writeable. Writing a zero to bit [24] of the Primary Expansion ROM Setup register disables this BAR. The minimum size for this address range is 4 KB. The maximum size is 16 MB. *Reset value:* 0 (disabled).

7.2.3.9 Downstream I/O or Memory 1 and Upstream I/O or Memory 0 Translated Base Register

Dsw

	Downstream I/O or Memory 1 Translated Base	Upstream I/O or Memory 0 Translated Base
Primary byte offset:	9B:98h	A7:A4h
Secondary byte offset:	9B:98h	A7:A4h
CSR byte offset:	06F:06Ch	07C:078h

These registers contain the translated base addresses for their respective downstream and upstream BARs. The base address of the transaction on the initiator bus is replaced by the base address contained in these registers. These registers are also mapped in the 21554 I/O and memory CSR space.

Table 7-33. Downstream I/O or Memory 1 and Upstream I/O or Memory 0 Translated Base Register

Bit	Name	R/W	Description
5:0	Reserved	R	Reserved. Returns 0 when read.
31:6	XLAT_BASE	R/W	Contains the translated base address for downstream or upstream transactions whose initiator bus addresses fall into either the Downstream I/O or Memory 1, or Upstream I/O or Memory 0 Base Address range. The number of bits that are used for the translated base is determined by the setup register corresponding to that base address and also matches the number of writeable bits in the corresponding BAR. The remaining bits may be written but are ignored when performing address translation. When an I/O or memory transaction is initiated by the 21554 on the target bus, the original base address is replaced with the value contained in this register.

7.2.3.10 Downstream Memory 0, 2, 3, and Upstream Memory 1 Translated Base Register

	Downstream Memory 0 Translated Base	Downstream Memory 2 Translated Base	Downstream Memory 3 Translated Base
Primary byte offset:	97:94h	9F:9Ch	A3:A0h
Secondary byte offset:	97:94h	9F:9Ch	A3:A0h
CSR byte offset:	06B:068h	073:070h	077:074h
	Upstream Memory 1 Translated Base		
Primary byte offset:	AB:A8h		
Secondary byte offset:	AB:A8h		
CSR byte offset:	07F:07Ch		

These registers contain the translated base addresses for their respective downstream and upstream BARs. The base address of the transaction on the initiator bus is replaced by the base address contained in these registers.

These registers are also mapped in the 21554 I/O and memory CSR space.

Table 7-34. Downstream Memory 0, 2, 3, and Upstream Memory 1 Translated Base Register

Bit	Name	R/W	Description
11:0	Reserved	R	Reserved. Returns 0 when read.
31:12	XLAT_BASE	R/W	Contains the translated base address for downstream or upstream transactions whose initiator bus addresses fall into one of the following address ranges: • Downstream Memory 0 (above low 4K boundary) • Downstream Memory 2 • Downstream Memory 3 • Upstream Memory 1 The number of bits that are used for the translated base is determined by the setup register corresponding to that base address and also matches the number of writeable bits in the corresponding BAR. The remaining bits can be written but are ignored when performing address translation. When a memory transaction is initiated by the 21554 on the target bus, the original base address is replaced with the value contained in this register.

7.2.3.11 Downstream I/O or Memory 1 and Upstream I/O or Memory 0 Setup Registers

	Downstream I/O or Memory 1 Setup	Upstream I/O or Memory 0 Setup
Primary byte offset:	B3:B0h	C7:C4h
Secondary byte offset:	B3:B0h	C7:C4h

These registers may be preloaded by serial ROM or programmed by the local processor before host configuration.

Table 7-35. Downstream I/O or Memory 1 and Upstream I/O or Memory 0 Setup Registers

Bit	Name	R/W	Description
0	Type Selector	R/(WS)	*When 0:* The base address register is requesting memory space, or is disabled. *When 1:* The base address register is requesting I/O space. *Reset value:* 0
2:1	Type	R/(WS)	Type of space requested. Allowable values: • 00b to indicate that the space requested by the BAR may be located anywhere in memory space, must be used for I/O space • 01b to indicate that memory space must be mapped below a 1MB boundary Other values may have unpredictable results. *Reset value:* 00h.
3	Prefetchable	R/(WS)	Indicates whether the space requested by the BAR is prefetchable. *When 0:* Not prefetchable (required, but not hardware enforced, for I/O space). *When 1:* Prefetchable. *Reset value*: 0
5:4	Reserved	R	Read only as 0.
30:6	Size	R/(WS)	These bits specify the size of the address range requested by the BAR. When a bit is 1, the corresponding bit in the BAR functions as a readable and writeable bit. When a bit is 0, the corresponding bit in the BAR functions as a read-only bit that always returns zero when read. The *PCI Local Bus Specification* states that the maximum value requested for I/O space should not be greater than 256 bytes, although this is not enforced in hardware. When configured as a memory range, bits [11:6] should be set to a 0 as the minimum supported memory range is 4KB. *Reset value:* 0 (disabled).
31	BAR_Enable	R/(WS)	Base Address Register enable. *When 0:* The corresponding BAR is disabled and reads as 0. *When 1:* The corresponding BAR is enabled, with size and type specified by this setup register. *Reset value:* 0

7.2.3.12 Downstream Memory 0, 2, 3, and Upstream Memory 1 Setup Registers

	Downstream Memory 0 Setup	Downstream Memory 2 Setup	Downstream Memory 3 Setup
Primary byte offset:	AF:ACh	B7:B4h	BB:B8h
Secondary byte offset:	AF:ACh	B7:B4h	BB:B8h
	Upstream Memory 1 Setup		
Primary byte offset:	CB:C8h		
Secondary byte offset:	CB:C8h		

These registers are used to program the type and size of their respective upstream and downstream base address registers.

Table 7-36. Downstream Memory 0, 2, 3, and Upstream Memory 1 Setup Registers

Bit	Name	R/W	Description
0	Type Selector	R	Read only as 0 to indicate memory space is requested by the corresponding memory base address register.
2:1	Type	R/(WS)	Type of space requested. Allowable values are: • 00b to indicate that the space requested by the base address register may be located anywhere in memory space • 01b to indicate that it must be mapped below a 1MB boundary • 10b for Downstream Memory 3 Setup register to request a 64-bit BAR Other values may yield unpredictable results. *Reset value:* 00b.
3	Prefetchable	R/(WS)	Indicates whether the space requested by the base address register is prefetchable. *When 0:* Not prefetchable. *When 1:* Prefetchable. *Reset value*: 0
11:4	Reserved	R	Read only as 0.
30:12	Size	R/(WS)	These bits specify the size of the address range requested by the base address register. When a bit is 1, the corresponding bit in the BAR functions as a readable and writeable bit. When a bit is 0, the corresponding bit in the BAR functions as a read-only bit that always returns zero when read. Note: If this field of the DS Memory 0 Setup is written to all zeros, the 21554 will actually return 7FFFFh on reads (request 4 K). *Reset value*: 0 (disabled), except for Downstream Memory 0 Setup register, whose reset value is 7FFFFh (request 4 KB).
31	BAR_Enable	R/(WS)	Base Address Register enable. If the Upper 32 Bits Downstream Memory 3 Setup register bit [31] is a 1, then the corresponding BAR is enabled as a 64-bit register, and this bit is part of the size field for the 64-bit BAR. *When 0:* The corresponding BAR is disabled and reads as 0, with the exception noted above. *When 1:* The corresponding BAR is enabled, with size and type specified by this setup register. *Reset value:* 0, except for Downstream Memory 0 Setup register, whose reset value is 1.

7.2.3.13 Upper 32 Bits Downstream Memory 3 Setup Register

Primary byte offset: BF:BCh
Secondary byte offset: BF:BCh

This register may be preloaded by serial ROM or programmed by the local processor before host configuration.

Table 7-37. Upper 32 Bits Downstream Memory 3 Setup Register

Bit	Name	R/W	Description
30:0	Size	R/(WS)	These bits specify upper 32 bits of the size of the address range requested by Downstream Memory 3 Base Address register. When a bit is 1, the corresponding bit in Downstream Memory 3 BAR functions as a readable and writeable bit. When a bit is 0, the corresponding bit in Downstream Memory 3 BAR functions as a read-only bit that always returns 0 when read. These bits must be set to a non-zero value only when bits [2:1] of Downstream Memory 3 BAR are set to 10b (this is not enforced in hardware). *Reset value:* 0
31	BAR_Enable	R/(WS)	64-bit Downstream Memory 3 BAR enable. *When 0:* The Downstream Memory 3 64-bit BAR is disabled (but may still be a 32-bit BAR). *When 1:* The Downstream Memory 3 BAR is enabled as a 64-bit BAR. *Reset value:* 0 (disabled)

7.2.3.14 Primary Expansion ROM Setup Register

Primary byte offset: C3:C0h
Secondary byte offset: C3:C0h

This register may be preloaded by serial ROM or programmed by the local processor before host configuration.

Table 7-38. Primary Expansion ROM Setup Register

Bit	Name	R/W	Description
11:0	Reserved	R	Reserved. Read only as 0.
23:12	Size	R/(WS)	These bits specify the size of the address range requested by the Primary Expansion ROM Base Address register. When a bit is 1, the corresponding bit in the Primary Expansion ROM Base Address register functions as a readable and writeable bit. When a bit is 0, the corresponding bit in the Primary Expansion ROM Base Address register functions as a read-only bit that always returns zero when read. *Reset value:* 0 (disabled).
24	BAR_Enable	R/(WS)	Base Address Register enable. Note the location of this bit in the preload sequence, as shown in Section 7.3 and Chapter 20. *When 0:* The Primary Expansion ROM BAR is disabled and reads as 0. *When 1:* The Primary Expansion ROM BAR is enabled, with size specified by this setup register. *Reset value:* 0
31:25	Reserved	R	Reserved. Returns 0 when read.

7.2.4 Configuration Transaction Generation Registers

This section contains descriptions of configuration transaction generation registers.

7.2.4.1 Downstream and Upstream Configuration Address Registers

	Downstream Configuration Address	Upstream Configuration Address
Primary byte offset:	83:80h	8B:88h (Reserved)
Secondary byte offset:	83:80h (Reserved)	8B:88h
CSR Space	003:000h	00B:008h

This description covers both the downstream and upstream versions of this register. These registers are also mapped in memory and I/O space.

Table 7-39. Downstream and Upstream Configuration Address Registers

Bit	Name	R/W	Description
31:0	CFG_ADDR (CA)	DCA: R/(WP) UCA: R/(WS)	This register contains the address for a configuration transaction to be generated on the target bus. The address is driven exactly as written in this register. This register should be written before the corresponding Downstream or Upstream Configuration Data register is accessed. Once the Downstream or Upstream Configuration Data register is accessed, the transaction is initiated on the secondary or primary bus, respectively. If the semaphore method is used, a master should not write to this register unless the master has successfully read a 0 from the Downstream or Upstream Configuration Own bit. The Downstream Configuration Address register cannot be written from the secondary interface. The Upstream Configuration Address register cannot be written from the primary interface.

7.2.4.2 Downstream Configuration Data and Upstream Configuration Data Registers

	Downstream Configuration Data	Upstream Configuration Data
Primary byte offset:	87:84h	8F:8Ch (Reserved)
Secondary byte offset:	87:84h (Reserved)	8F:8Ch
CSR Space	007:004h	00F:00Ch

This description covers both the downstream and upstream versions of this register. These registers are also mapped in memory and I/O space. This register is treated as a reserved register for all memory accesses.

Table 7-40. Downstream Configuration Data and Upstream Configuration Data Registers

Bit	Name	R/W	Description
31:0	CFG_DATA (CD)	DCD: R/(WP) UCD: R/(WS)	This register contains the write data driven or the read data returned from a configuration transaction initiated by the 21554. The Downstream or Upstream Configuration Address register contains the address for this transaction, depending on the direction of the transaction. The transaction is initiated when this register is written (for a configuration write) or read (for a configuration read) and the corresponding Configuration Control bit is a one. The byte enables used for this register access are the same byte enables used for the transaction driven on the target bus. A target retry is returned to the initiator until the transaction has been completed on the target bus. If the semaphore method is used, a master should not write to this register unless the master has successfully read a 0 from the Downstream or Upstream Configuration Own bit. The Downstream Configuration Data register is reserved when accessed from the secondary interface, or on either interface when the Downstream Configuration Enable bit is not set. The Upstream Configuration Data register is reserved when accessed from the primary interface, or on either interface when the Upstream Configuration Enable bit is not set.

7.2.4.3 Configuration Own Bits Register

Primary byte offset: 91:90h
Secondary byte offset: 91:90h
CSR byte offset 011:010h.

This register is also mapped in memory and I/O space.

Table 7-41. Configuration Own Bits Register

Bit	Name	R/W	Description
0	Downstream Configuration Own Bit	R0TS(P) R(S)	Indicates ownership of the Downstream Configuration Address and Downstream Configuration Data registers. *When 0:* Downstream Configuration Address and Downstream Configuration Data registers are not owned. When read as a 0 from the primary interface, this bit is subsequently set to a 1 by the 21554 if the Downstream Configuration Control bit is a 1. *When 1:* A master owns Downstream Configuration Address and Downstream Configuration Data registers. If this semaphore method is used, other masters should not attempt to access these registers when this bit is a 1. This bit is automatically cleared once the configuration transaction has completed on the initiator bus. *Reset value:* 0.
7:1	Reserved	R	Read only as 0.
8	Upstream Configuration Own Bit	R0TS (S) R(P)	Indicates ownership of the Upstream Configuration Address and Upstream Configuration Data registers. *When 0:* Upstream Configuration Address and Upstream Configuration Data registers are not owned. When read as a 0 from the secondary interface, this bit is subsequently set to a 1 by the 21554 if the Upstream Configuration Control bit is a 1. *When 1:* A master owns Upstream Configuration Address and Upstream Configuration Data registers. If this semaphore method is used, other masters should not attempt to access these registers when this bit is a 1. This bit is automatically cleared once the configuration transaction has completed on the initiator bus. *Reset value:* 0.
15:9	Reserved	R	Read only as 0.

7.2.4.4 Configuration Control and Status Register

Primary byte offset: 93:92h
Secondary byte offset: 93:92h
CSR byte offset: 013:012h

This register is also mapped in memory and I/O space.

Table 7-42. Configuration Control and Status Register

Bit	Name	R/W	Description
0	Downstream Configuration Own Status	R	Provides the current value of the Downstream Configuration Own bit. This bit has no side effects when read.
1	Downstream Configuration Control	R/W	Enables the 21554 to perform downstream indirect configuration transactions. *When 0:* The 21554 will not initiate a configuration transaction on the secondary interface when the Downstream Configuration Data register is accessed. The Downstream Configuration Data register is treated as a reserved register. *When 1:* The 21554 is enabled to perform downstream configuration transactions when the Downstream Configuration Data register is accessed. *Reset value:* 0
7:2	Reserved	R	Reserved. Returns 0 when read.
8	Upstream Configuration Own Status	R	Provides the current value of the Upstream Configuration Own bit. This bit has no side effects when read.
9	Upstream Configuration Control	R/W	Enables the 21554 to perform upstream indirect configuration transactions. *When 0:* The 21554 will not initiate a configuration transaction on the primary interface when the Upstream Configuration Data register is accessed. The Upstream Configuration Data register is treated as a reserved register. *When 1:* The 21554 is enabled to perform upstream configuration transactions when the Upstream Configuration Data register is accessed. *Reset value:* 0
15:10	Reserved	R	Reserved. Reads only as 0.

7.2.5 Device-Specific Control and Status Registers

This section contains information about the device-specific control and status registers.

7.2.5.1 Chip Control 0 Register

Primary byte offset: CD:CCh
Secondary byte offset: CD:CCh

This register may be preloaded by serial ROM or programmed by the local processor before host configuration.

Table 7-43. Chip Control 0 Register (Sheet 1 of 3)

Bit	Name	R/W	Description
0	Master Abort Mode	R/W	Controls the 21554's behavior on the initiator bus when a master abort termination occurs in response to a delayed transaction initiated by the 21554 on the target bus. *When 0:* The 21554 asserts TRDY# in response to a delayed transaction, and returns FFFFFFFFh if a read. *When 1:* The 21554 returns a target abort in response to a delayed transaction. *Reset value:* 0
1	Memory Write Disconnect Control	R/W	Controls the disconnect boundary for memory writes. This bit does not apply to memory write and invalidate commands. *When 0:* The 21554 disconnects memory writes either on an aligned 4KB boundary, a page boundary (Upstream Memory Range 2 only) or when the posted write queue is full. *When 1:* The 21554 disconnects memory write on an aligned cache line boundary, or when the posted write queue is full. *Reset value:* 0
2	Primary Master Timeout	R/W	Sets the maximum number of PCI clock cycles that the 21554 waits for an initiator on the primary bus to repeat a delayed transaction request. The counter starts when the delayed transaction completion is ready to be returned to the initiator. If the initiator has not repeated the transaction at least once before the counter expires, the 21554 discards the delayed transaction from its queues. *When 0:* The primary master timeout counter is 2^{15} PCI clock cycles, or .983ms for a 33-MHz bus. *When 1:* The value is 2^{10} PCI clock cycles, or 30.7 µs for a 33-MHz bus. *Reset Value:* 0
3	Secondary Master Timeout	R/W	Sets the maximum number of PCI clock cycles that the 21554 waits for an initiator on the secondary bus to repeat a delayed transaction request. The counter starts when the delayed transaction completion is ready to be returned to the initiator. If the initiator has not repeated the transaction at least once before the counter expires, the 21554 discards the delayed transaction from its queues. *When 0:* The secondary master timeout counter is 2^{15} PCI clock cycles, or .983ms for a 33-MHz bus. *When 1:* The value is 2^{10} PCI clock cycles, or 30.7 µs for a 33-MHz bus. *Reset Value:* 0

Table 7-43. Chip Control 0 Register (Sheet 2 of 3)

Bit	Name	R/W	Description
4	Primary Master Timeout Disable	R/W	Disables the primary master timeout counter. *When 0:* The primary master timeout counter is enabled and uses the value specified by the Primary Master timeout bit. *When 1:* The primary master timeout counter is disabled. The 21554 waits indefinitely for a primary bus initiator to repeat a delayed transaction. *Reset value:* 0
5	Secondary Master Timeout Disable	R/W	Disables the secondary master timeout counter. *When 0:* The secondary master timeout counter is enabled and uses the value specified by the Secondary Master Timeout bit. *When 1:* The secondary master timeout counter is disabled. The 21554 waits indefinitely for a secondary bus initiator to repeat a delayed transaction. *Reset value:* 0
6	Delayed Transaction Order Control	R/W	Controls how the 21554 initiates delayed transactions on the target bus. *When 0:* The 21554 uses a round-robin arbitration scheme to determine which transaction is attempted. After receiving a target retry in response to a delayed transaction, the 21554 can initiate a different queued delayed transaction. *When 1:* When a target retry is received in response to a delayed transaction, the 21554 continues to attempt that same transaction until a response other than target retry is received. The 21554 does not initiate other delayed transactions until the above condition is satisfied. *Reset value: 0.*
7	SERR# Forward Enable	R/W	SERR# forward enable. *When 0:* The 21554 does not assert p_serr_l as a result of s_serr_l assertion. *When 1:* The 21554 asserts p_serr_l whenever s_serr_l is detected asserted and the primary SERR# Enable bit is set. *Reset value:* 0
8	Upstream DAC Prefetch Disable	R/W	Controls prefetching for upstream dual address transactions using the memory read bus command. *When 0:* Prefetching is performed for upstream DAC memory reads. *When 1:* Upstream DACs using the memory read bus command are not prefetched; transactions are limited to a single Dword and byte enables are preserved. *Reset value:* 0
9	Multiple Device Enable	R/W	Enables multiple devices to be attached to the ROM interface. *When 0:* Only the parallel and serial ROM can be attached to the ROM interface. The parallel ROM chip select is driven on the pr_cs_l pin. *When 1:* Multiple devices may be attached to the ROM interface. All chip selects with the exception of the serial ROM are decoded from the upper address lines of the ROM interface. *Reset value:* 0

Table 7-43. Chip Control 0 Register (Sheet 3 of 3)

Bit	Name	R/W	Description
10	Primary Access Lockout	R/(WS)	This bit prevents the primary bus from accessing configuration space. This allows the local processor to access the 21554 registers before the host processor accesses them. This bit can be written from the secondary interface only. The local processor must write this bit to a 0 to allow the 21554 to be configured by the host processor, unless preloaded to 0 by serial ROM. *When 0:* The 21554 configuration space can be accessed from both interfaces. *When 1:* The 21554 configuration space can only be accessed from the secondary interface. Primary bus accesses, with the exception of the Reset Control register, receive a target retry. *Reset value:* 1 if pr_ad[3] is high during reset, 0 if pr_ad[3] is low during reset.
11	Secondary Clock Disable	R/W	Secondary clock output disable. *When 0:* s_clk_o is driven as a buffered copy of p_clk. *When 1:* s_clk_o is disabled and driven low. *Reset value:* 0 if pr_ad[5] is high during primary bus reset; 1 if pr_ad[5] is low during primary bus reset.
13:12	Reserved	R	Reserved. Read only as 00b.
15:14	VGA Mode	R/W	Enables address decoding and transaction forwarding of the following VGA transactions: • Frame buffer memory addresses 000BFFFF:000A0000h • VGA I/O addresses 3BB:3B0h and 3DF:3C0h, where AD[31:16] = 0000h and AD[15:10] are not decoded. The following values control how the 21554 decodes and forwards VGA memory and I/O transactions: • 00: VGA memory and I/O transactions on the primary and secondary buses are ignored (unless decoded by some other mechanism). • 01: VGA memory and I/O transactions on the primary bus are forwarded to the secondary bus. VGA transactions on the secondary bus are ignored. • 10: VGA memory and I/O transactions on the secondary bus are forwarded to the primary bus. VGA transactions on the primary bus are ignored. • 11: Illegal.The 21554 behavior is unpredictable. *Reset value*: 00b

7.2.5.2 Chip Control 1 Register

Primary byte offset: CF:CEh
Secondary byte offset: CF:CEh

This register may be preloaded by serial ROM or programmed by the local processor before host configuration.

Table 7-44. Chip Control 1 Register (Sheet 1 of 2)

Bit	Name	R/W	Description
0	Primary Posted Write Threshold	R/W	Controls the queue full threshold limit of the downstream posted write queue. When the queue is designated full, the 21554 returns retry to posted writes on the primary bus. Otherwise, the 21554 accepts write data into the posted write queue. *When 0:* Posted write queue full when less than a cache line is free to post data. *When 1:* Posted write queue full when less than a half cache line (for CLS=8,16,32) is free to post data. If CLS=4, a full cache line threshold is used. *Reset value:* 0b
1	Secondary Posted Write Threshold	R/W	Controls the queue full threshold limit of the upstream posted write queue. When the queue is designated full, the 21554 returns retry to posted writes on the secondary bus. Otherwise, the 21554 accepts write data into the posted write queue. *When 0:* Posted write queue full when less than a cache line is free to post data. *When 1:* Posted write queue full when less than a half cache line (for CLS=8,16,32) is free to post data. If CLS=4, a full cache line threshold is used. *Reset value:* 0b
3:2	Primary Delayed Read Threshold	R/W	Controls the read data queue threshold for initiating read transactions on the primary bus. When the amount of read data in the queue exceeds the threshold, the 21554 does not initiate a pending upstream delayed memory read transaction on the primary bus. The following values control when the 21554 initiates a memory read: • 00b: At least eight Dwords free in read data queue for all memory read commands • Reserved • 10b: At least one cache line free for MRL and MRM, eight Dwords free for memory read • 11b: At least one cache line free for all memory read commands Note that the secondary bus cache line size is used for the threshold calculation. *Reset value:* 00b
5:4	Secondary Delayed Read Threshold	R/W	Controls the read data queue threshold for initiating read transactions on the secondary bus. When the amount of read data in the queue exceeds the threshold, the 21554 does not initiate a pending downstream delayed memory read transaction on the secondary bus. The following values control when the 21554 initiates a memory read: • 00b: At least eight Dwords free in read data queue for all memory read commands • Reserved • 10b: At least one cache line free for MRL and MRM, eight Dwords free for memory read • 11b: At least one cache line free for all memory read commands Note that the primary bus cache line size is used for the threshold calculation. *Reset value:* 00b

Table 7-44. Chip Control 1 Register (Sheet 2 of 2)

Bit	Name	R/W	Description
7:6	Subtractive Decode Enable	R/W	Controls subtractive decoding for downstream and upstream I/O transactions. When the 21554 is enabled to perform subtractive decoding in one direction, those transactions are forwarded to the opposite bus with no address translation. Possible values are: • 00: No subtractive decoding is performed on either interface. • 01: I/O subtractive decoding enabled on primary interface. • 10: I/O subtractive decoding enabled on secondary interface. • 11: Illegal. Results are unpredictable. *Reset value:* 00
11:8	Page Size	R/W	Selects the page size used for the Upstream Memory 2 address range. The total size of this range is dependent on the page size. Possible page size values and their encoding are: • 0h : Disables the Upstream Memory 2 Base address register. • 1h : 256 bytes • 2h : 512 bytes • 3h : 1 KB • 4h : 2 KB • 5h : 4 KB • 6h : 8 KB • 7h : 16 KB • 8h : 32 KB • 9h : 64 KB • Ah : 128 KB • Bh : 256 KB • Ch : 512 KB • Dh : 1 MB • Eh : 2 MB • Fh : 4 MB Reset value: 0h
12	I_2O_ENA	R/W	Enables the I_2O message unit. *When 0:* The I_2O message unit is disabled. Memory accesses to the Inbound and Outbound FIFO registers at CSR offsets 40h and 44h result in TRDY# and discarded data on writes, and TRDY# with a return of FFFFFFFFh on reads. *When 1:* The I_2O message unit is enabled. Memory writes cause a posting to the Inbound Post or Outbound Free list; Reads remove an entry from the Inbound Free or Outbound Post list. *Reset value:* 0.
15:13	I_2O_SIZE	R/W	Selects the I_2O FIFO size. The 21554 supports the following values: • 000b: 256 entries • 001b: 512 entries • 010b: 1 K entries • 011b: 2 K entries • 100b: 4 K entries • 101b: 8 K entries • 110b: 16 K entries • 111b: 32 K entries *Reset value:* 000b

7.2.5.3 Reset Control Register

Primary byte offset: DB:D8h
Secondary byte offset: DB:D8h

This register is accessible from the primary interface regardless of the state of the Primary Lockout bit.

Table 7-45. Reset Control Register

Bit	Name	R/W	Description
0	Secondary Reset	R/(WP)	Secondary bus reset. *When 0:* The 21554 deasserts s_rst_l. This bit must be cleared by a configuration write in the case when it is set by a configuration write. Otherwise, it clears automatically after 100 µsec or when p_rst_l deasserts. *When 1*: The 21554 asserts s_rst_l. This bit is set automatically when the Chip Reset bit is written with a 1 or when p_rst_l is asserted, or is set with a configuration write. *Reset value:* 0 (disabled).
1	Chip Reset	R/(WP)	Chip reset control. *When 1:* Causes the 21554 to perform a chip reset and to assert s_rst_l. Data buffers, configuration registers, and both the primary and secondary interfaces are reset to their initial state. The 21554 clears this bit once chip reset is complete. *Reset value:* 0
2	Subsystem Status	R	Reflects the state of the s_pme_l pin. If not used for power management, the s_pme_l pin may be used to indicate local subsystem status. *When 0:* s_pme_l is at a deasserted high level. *When 1:* s_pme_l is at an asserted low level.
3	**l_stat** Status	R	Reflects the state of the l_stat pin. *When 0:* l_stat is low. *When 1:* l_stat is high.
31:3	Reserved	R	Reserved. Reads only as 0.

7.2.5.4 Chip Status Register

Primary byte offset: D1:D0h
Secondary byte offset: D1:D0h

All of the following conditions can cause the assertion of p_serr_l or s_serr_l if the corresponding SERR# enable bit is set and the disable bit for this condition is not set.

Table 7-46. Chip Status Register

Bit	Name	R/W	Description
0	Downstream Delayed Transaction Master Time-out	R/W1TC	This bit is set to a 1 and p_serr_l is conditionally asserted when the primary master timeout counter expires and a downstream delayed transaction completion is discarded from the 21554's queues. *Reset value:* 0
1	Downstream Delayed Read Transaction Discarded	R/W1TC	This bit is set to a 1 and p_serr_l is conditionally asserted when the 21554 discards a downstream delayed read transaction request after receiving 2^{24} target retries from the secondary bus target. *Reset value:* 0
2	Downstream Delayed Write Transaction Discarded	R/W1TC	This bit is set to a 1 and p_serr_l is conditionally asserted when the 21554 discards a downstream delayed write transaction request after receiving 2^{24} target retries from the secondary bus target. *Reset value:* 0
3	Downstream Posted Write Data Discarded	R/W1TC	This bit is set to a 1 and p_serr_l is conditionally asserted when the 21554 discards a downstream posted write transaction after receiving 2^{24} target retries from the secondary bus target. Reset value: 0
7:4	Reserved	R	Reserved. Returns 0 when read.
8	Upstream Delayed Transaction Master Time-out	R/W1TC	This bit is set to a 1 and s_serr_l is conditionally asserted when the secondary master timeout counter expires and an upstream delayed transaction completion is discarded from the 21554's queues. *Reset value:* 0
9	Upstream Delayed Read Transaction Discarded	R/W1TC	This bit is set to a 1 and s_serr_l is conditionally asserted when the 21554 discards an upstream delayed read transaction request after receiving 2^{24} target retries from the primary bus target. *Reset value:* 0
10	Upstream Delayed Write Transaction Discarded	R/W1TC	This bit is set to a 1 and s_serr_l is conditionally asserted when the 21554 discards an upstream delayed write transaction request after receiving 2^{24} target retries from the primary bus target. *Reset value:* 0
11	Upstream Posted Write Data Discarded	R/W1TC	This bit is set to a 1 and s_serr_l is conditionally asserted when the 21554 discards an upstream posted write transaction after receiving 2^{24} target retries from the primary bus target. *Reset value:* 0
15:12	Reserved	R	Reserved. Returns 0 when read.

7.2.5.5 Arbiter Control Register

Primary byte offset: D3:D2h
Secondary byte offset: D3:D2h

This register may be preloaded by serial ROM or programmed by local processor before host configuration.

Table 7-47. Arbiter Control Register

Bit	Name	R/W	Description
9:0	Arbiter Control	R/W	Each bit controls whether a secondary bus master is assigned to the high priority arbiter ring or the low priority arbiter ring. Bits [8:0] correspond to request inputs s_req_l[8:0], respectively. Bit [9] corresponds to the internal 21554 secondary bus request. *When 0:* Indicates that the master belongs to the low priority group. *When 1:* Indicates that the master belongs to the high priority group. *Reset value:* 10 0000 0000b.
15:10	Reserved	R	Reserved. Returns 0 when read.

7.2.5.6 Primary SERR# Disable Register

Primary byte offset: D4h
Secondary byte offset: D4h

This register may be preloaded by serial ROM or programmed by the local processor before host configuration. This register controls the ability of the 21554 to assert p_serr_l for a particular condition. When the bit is a 0, the assertion of p_serr_l is not masked for this event. When the bit is a 1, the assertion of p_serr_l is masked for this event.

Table 7-48. Primary SERR# Disable Register

Bit	Name	R/W	Description
0	Downstream Delayed Transaction Master Time-out	R/W	Disables p_serr_l assertion when a downstream master time-out condition is detected and the downstream transaction is discarded. *Reset value:* 0
1	Downstream Delayed Read Transaction Discarded	R/W	Disables p_serr_l assertion when 21554 discards a downstream delayed read transaction request after receiving 2^{24} target retries from secondary bus target. *Reset value:* 0
2	Downstream Delayed Write Transaction Discarded	R/W	Disables p_serr_l assertion when 21554 discards a downstream delayed write transaction request after receiving 2^{24} target retries from secondary bus target. *Reset value:* 0
3	Downstream Posted Write Data Discarded	R/W	Disables p_serr_l assertion when 21554 discards a downstream posted write transaction after receiving 2^{24} target retries from secondary bus target. *Reset value:* 0
4	Target Abort during Downstream Posted Write	R/W	Disables p_serr_l assertion when 21554 detects a target abort on the secondary interface in response to a downstream posted write. *Reset value:* 0
5	Master Abort during Downstream Posted Write	R/W	Disables p_serr_l assertion when the 21554 detects a master abort on the secondary interface when initiating a downstream posted write. *Reset value:* 0
6	Downstream Posted Write Parity Error	R/W	Disables p_serr_l assertion when the 21554 detects s_perr_l asserted during a downstream posted write. *Reset value:* 0
7	Reserved	R	Reserved. Returns 0 when read.

7.2.5.7 Secondary SERR# Disable Register

Primary byte offset: D5h
Secondary byte offset: D5h

This register may be preloaded by serial ROM or programmed by the local processor before host configuration. This register controls the ability of the 21554 to assert s_serr_l for a particular condition. When the bit is a 0, the assertion of s_serr_l is not masked for this event. When the bit is a 1, the assertion of s_serr_l is masked for this event.

Table 7-49. Secondary SERR# Disable Register

Bit	Name	R/W	Description
0	Upstream Delayed Transaction Master Timeout	R/W	Disables s_serr_l assertion when an upstream master timeout condition is detected and the upstream transaction is discarded. *Reset value:* 0
1	Upstream Delayed Read Transaction Discarded	R/W	Disables s_serr_l assertion when the 21554 discards an upstream delayed read transaction request after receiving 2^{24} target retries from the primary bus target. *Reset value:* 0
2	Upstream Delayed Write Transaction Discarded	R/W	Disables s_serr_l assertion when the 21554 discards an upstream delayed write transaction request after receiving 2^{24} target retries from the primary bus target. *Reset value:* 0
3	Upstream Posted Write Data Discarded	R/W	Disables s_serr_l assertion when the 21554 discards an upstream posted write transaction after receiving 2^{24} target retries from the primary bus target. *Reset value:* 0
4	Target Abort during Upstream Posted Write	R/W	Disables s_serr_l assertion when the 21554 detects a target abort on the primary interface in response to an upstream posted write. *Reset value:* 0
5	Master Abort during Upstream Posted Write	R/W	Disables s_serr_l assertion when the 21554 detects a master abort on the primary interface when initiating an upstream posted write. *Reset value:* 0
6	Upstream Posted Write Parity Error	R/W	Disables s_serr_l assertion when the 21554 detects p_perr_l asserted during an upstream posted write. *Reset value:* 0
7	Reserved	R	Reserved. Returns 0 when read.

7.3 Serial ROM Configuration Data Preload Format

Some fields of the 21554 configuration registers may be preloaded using the serial ROM interface. The first two bits read from the serial ROM after the completion of chip reset indicates whether a register preload should be performed. If the first two bits read as 10b, then an auto-load sequence is initiated. The ROM is sequentially read and the data is shifted along a scan register chain to load the registers listed in Table 7-50. For a detailed, bit-by-bit summary of the preload sequence, see Chapter 20.

Table 7-50. Autoload Sequence

Register	Description
byte 00h	Register preload control Bits [7:6] must read as 10b to enable the register preload; otherwise the serial ROM read is terminated and registers remain at their reset values. Bits [5:0] are reserved and must be 0.
bytes 03:01h	Reserved
bytes 04h	Primary Programming Interface
byte 05h	Primary Sub-Class Code
byte 06h	Primary Base Class Code
byte 08:07h	Subsystem Vendor ID Register
byte 0A:09h	Subsystem ID Register
byte 0Bh	Primary Minimum Grant Register
byte 0Ch	Primary Maximum Latency Register
byte 0Dh	Secondary Programming Interface Register
byte 0Eh	Secondary Sub-Class Code Register
byte0Fh	Secondary Base Class Code Register
byte 10h	Secondary Minimum Grant Register
byte 11h	Secondary Maximum Latency Register
byte 15:12h	Downstream Memory 0 Setup Register
byte 19:16	Downstream I/O or Memory 1 Setup Register
byte 1D:1Ah	Downstream Memory 2 Setup Register
byte 21:1Eh	Downstream Memory 3 Setup Register
byte 25:22h	Downstream Memory 3 Setup Upper 32 Bits Register
byte 27:26h	Primary Expansion ROM Setup Register
byte 2B:28h	Upstream I/O or Memory 0 Setup Register
byte 2F:2Ch	Upstream Memory 1 Setup Register
byte 31:30h	Chip Control 0 Register
byte 33:32h	Chip Control 1 Register
byte 35:34h	Arbiter Control Register
byte 36h	Primary SERR# Disables Register
byte 37h	Secondary SERR# Disables Register
byte 3F:38h	PCI Power Management Data Register Values
byte 40h	Reserved
byte 42:41h	PCI Power Management and BiST

Bytes 42:41h containing PCI Power Management setup information and BiST control have the format shown in Table 7-51.

Table 7-51. Power Management and BiST Autoload Format

Register	Description
Bit 15:10	PCI Power Management Capabilities Register [14:9]
Bit 9	PCI Power Management Capabilities Register [5]
Bit 8:6	PCI Power Management Capabilities Register [2:0]
Bit 5:4	PCI Power Management Control and Status Register [14:13]
Bit 3	PCI Power Management Data Register Enable
Bit 2	BiST Register [7]

Control and Status Registers 8

The 21554 implements a number of registers that may be mapped into memory space on the primary interface using the Primary CSR Memory BAR or on the secondary interface using the Secondary CSR Memory BAR. These registers may also be mapped into I/O space on the primary interface using the Primary CSR I/O BAR or on the secondary interface using the Secondary CSR I/O BAR. The Address Translation Lookup Table is mapped directly into memory space requested by the primary and secondary BARs, but can only be accessed indirectly in I/O space.

All locations (with the exception of the Lookup Table) are byte accessible during memory and I/O writes. Reads of the registers are disconnected after one data phase.

Read accesses of reserved or unimplemented registers complete normally and return a data value of zero when read. Writes to reserved registers are treated as no-ops. That is, the write access is completed normally and the write data discarded.

Software must be careful when accessing registers that have bit fields reserved for future use. For read accesses, software must use appropriate masks to extract the defined bits and may not rely on reserved bits being any particular value. For write accesses, software must ensure that the values of reserved bit positions are preserved. That is, the values of the reserved bit positions must first be read, merged with the new values for other bit positions, and the merged data then written back.

8.1 CSR Address Map

This section shows the 21554 CSR register map. Figure 8-1 shows the registers and byte offsets. All registers are accessible in memory space. Registers are accessible in I/O space as designated in the table. If indicated as reserved — “res” shown in I/O space — reads to these registers through I/O space return 0, and writes have no effect. If indicated as not supported — “n” shown in I/O space — then the 21554 does not respond to I/O accesses to that address.

Figure 8-1. CSR Register Map (Sheet 1 of 2)

Byte 3	Byte 2	Byte 1	Byte 0	Byte Offset	I/O Space
Downstream Configuration Address[1]				000h	y
Downstream Configuration Data[1]				004h	y
Upstream Configuration Address[1]				008h	y
Upstream Configuration Data[1]				00Ch	y
Configuration Control and Status[1]		Configuration Own Bits[1]		010h	y
Downstream I/O Address				014h	y
Downstream I/O Data				018h	y
Upstream I/O Address				01Ch	y
Upstream I/O Data				020h	y
I/O Control and Status		I/O Own Bits		024h	y
Reserved			Lookup Table Offset	028h	y
Lookup Table Data				02Ch	y
I_2O Outbound Post_List Status				030h	y
I_2O Outbound Post_List Interrupt Mask				034h	y
I_2O Inbound Post_List Status				038h	y
I_2O Inbound Post_List Interrupt Mask				03Ch	y
I_2O Inbound Queue				040h	res
I_2O Outbound Queue				044h	res
I_2O Inbound Free_List Head Pointer				048h	y
I_2O Inbound Post_List Tail Pointer				04Ch	y
I_2O Outbound Free_List Tail Pointer				050h	y
I_2O Outbound Post_List Head Pointer				054h	y
I_2O Inbound Post_List Counter				058h	res
I_2O Inbound Free_List Counter				05Ch	res
I_2O Outbound Post_List Counter				060h	res
I_2O Outbound Free_List Counter				064h	res
Downstream Memory 0 Translated Base				068h	y
Downstream I/O or Memory 1 Translated Base				06Ch	y
Downstream Memory 2 Translated Base				070h	y
Downstream Memory 3 Translated Base				074h	y

1. Also mapped in 21554 configuration space.

FM-06137.AI4

Figure 8-2. CSR Register Map (Sheet 2 of 2)

Byte 3	Byte 2	Byte 1	Byte 0	Byte Offset	I/O Space
Upstream I/O or Memory 0 Translated Base				078h	y
Upstream Memory 1 Translated Base				07Ch	y
Chip Status CSR		Reserved		080h	y
Chip Clear IRQ Mask Register		Chip Set IRQ Mask Register		084h	y
Upstream Page Boundary IRQ 0 Register				088h	y
Upstream Page Boundary IRQ 1 Register				08Ch	y
Upstream Page Boundary IRQ Mask 0 Register				090h	y
Upstream Page Boundary IRQ Mask 1 Register				094h	y
Secondary Clear IRQ		Primary Clear IRQ		098h	y
Secondary Set IRQ		Primary Set IRQ		09Ch	y
Secondary Clear IRQ Mask		Primary Clear IRQ Mask		0A0h	y
Secondary Set IRQ Mask		PrimarySet IRQ Mask		0A4h	y
Scratchpad 0				0A8h	y
Scratchpad 1				0ACh	y
Scratchpad 2				0B0h	y
Scratchpad 3				0B4h	y
Scratchpad 4				0B8h	y
Scratchpad 5				0BCh	y
Scratchpad 6				0C0h	y
Scratchpad 7				0C4h	y
Reserved	ROM Data	ROM Setup		0C8h	y
ROM Control	ROM Address			0CCh	y
Reserved				0D0h-0FFh	y
Upstream Memory 2 Lookup Table				100h-1FFh	n
Reserved				200h-FFFh	n

1. Also mapped in 21554 configuration space.

FM-06232.AI4

8.2 CSR Descriptions

This section contains the individual register descriptions of the 21554 CSR space. This section is divided into the following subsections:

- I/O and Configuration Transaction Generation Registers
- I_2O Registers
- Address Registers
- Interrupt Registers
- ROM Registers
- Control and Miscellaneous Registers

Registers that perform the same function for primary and secondary interfaces are described together.

8.2.1 I/O and Configuration Transaction Generation Registers

This section contains descriptions of the I/O and Configuration Transaction Generation Registers.

8.2.1.1 Downstream /Upstream Configuration Address and Data, Control, Status Registers

All of these registers are mapped into the 21554 configuration space and described in Section 7.2.4. Note that the 21554 initiates a transaction only when the Configuration Data registers are accessed at these locations using I/O reads and writes.

Downstream Configuration Address Register Byte Offset:	003:000h
Downstream Configuration Data Register Byte Offset:	007:004h
Upstream Configuration Address Register Byte Offset:	00B:008h
Upstream Configuration Data Register Byte Offset:	00F:00Ch
Configuration Own Bits Register Byte Offset:	011:010h
Configuration Control and Status Register Byte Offset:	013:012h

The Downstream Configuration Data Register and the Upstream Configuration Data Register are treated as reserved registers for all memory accesses.

8.2.1.2 Downstream I/O Address and Upstream I/O Address Registers

	Downstream I/O Address	Upstream I/O Address
Byte Offset:	017:014h	1F:1Ch

The Downstream I/O Address register is used for I/O transactions to be initiated on the secondary bus, and the Upstream I/O Address register is used for I/O transactions to be initiated on the primary bus. The downstream register can be written from the primary interface only and the upstream register can be written from the secondary interface only.

Table 8-1. Downstream I/O Address and Upstream I/O Address Registers

Bit	Name	R/W	Description
31:0	IO_ADDR (IA)	DIA: R/(WP) UIA: R/(WS)	This register contains the address for an I/O transaction to be generated on the target bus. The address is driven exactly as written in this register. This register should be written before the Downstream or Upstream I/O Data register is accessed. Once the Downstream or Upstream I/O Data register is written or read, the transaction is initiated on the secondary bus. If the semaphore method is used, a master should not write to this register unless the master has successfully read a 0 from the Downstream or Upstream I/O Own bit. *Reset value:* 0

8.2.1.3 Downstream I/O Data and Upstream I/O Data Registers

	Downstream I/O Data	Upstream I/O Data
Byte Offset:	01B:018h	023:020h

The Downstream I/O Data register is used for I/O transactions to be initiated on the secondary bus, and the Upstream I/O Data register is used for I/O transactions to be initiated on the primary bus. A downstream transaction is initiated by a primary interface I/O register access only and an upstream transaction is initiated by a secondary interface I/O access only.

Table 8-2. Downstream I/O Data and Upstream I/O Data Registers

Bit	Name	R/W	Description
31:0	IO_DATA (ID)	DID: R/(WP) UID: R/(WS) Res (Mem)	This register contains the write data driven or the read data returned from an I/O transaction initiated on the target bus. The Downstream or Upstream I/O Address register contains the address for this transaction. The transaction is initiated when this register is written (for an I/O write) or read (for an I/O read) using an I/O transaction. This register is treated as a reserved register for all memory accesses. The byte enables used for this register access are the same byte enables used for the transaction driven on the target bus. A target retry is returned to the initiator until the transaction has been completed on the target bus. *Reset value:* 0

8.2.1.4 I/O Own Bits Registers

Byte Offset: 025:024h

Table 8-3. I/O Own Bits Registers

Bit	Name	R/W	Description
0	Downstream I/O Own Bit	R0TS (P) R (S)	Indicates ownership of the Downstream I/O Address and Downstream I/O Data registers. *When 0:* Downstream I/O Address and Downstream I/O Data registers are not owned. When read as a 0 from the primary interface, this bit is subsequently set to a 1 by the 21554. *When 1:* Downstream I/O Address and Downstream I/O Data registers are owned by a master. If the semaphore method is used, other masters should not attempt to access these registers when this bit is a 1. This bit is automatically cleared once the I/O transaction has completed on the initiator bus. *Reset value:* 0.
7:1	Reserved	R	Read only as 0.
8	Upstream I/O Own Bit	R0TS (S) R (P)	Indicates ownership of the Upstream I/O Address and Upstream I/O Data registers. *When 0:* Upstream I/O Address and Upstream I/O Data registers are not owned. When read as a 0 from the secondary interface, this bit is subsequently set to a 1 by the 21554. *When 1:* Upstream I/O Address and Upstream I/O Data registers are owned by a master. If the semaphore method is used, other masters should not attempt to access these registers when this bit is a 1. This bit is automatically cleared once the I/O transaction has completed on the initiator bus. *Reset value:* 0.
15:9	Reserved	R	Read only as 0.

8.2.1.5 I/O Control and Status Register

Byte Offset: 027:026h

Table 8-4. I/O Control and Status Register

Bit	Name	R/W	Description
0	Downstream I/O Own Bit Status	R	This bit reflects the status of the Secondary Own bit used for generating I/O transaction on the secondary bus. *When 0:* the Downstream I/O Address and Downstream I/O Data registers are not owned. *When 1:* the Downstream I/O Address and Downstream I/O Data registers are owned by a master.
1	Downstream I/O Control	R/W	Enables the 21554 to perform downstream indirect I/O transactions. *When 0:* The 21554 will not initiate a I/O transaction on the secondary interface when the Downstream I/O Data register is accessed. The Downstream I/O Data register is treated as a reserved register. *When 1:* The 21554 is enabled to perform downstream I/O transactions when the Downstream I/O Data register is accessed with an I/O transaction. *Reset value:* 0
7:2	Reserved	R	Reserved. Read only as 0.
8	Upstream I/O Own Bit Status	R	This bit reflects the status of the Primary Own bit used for generating I/O transaction on the Primary bus. *When 0:* The Upstream I/O Address and Upstream I/O Data registers are not owned. *When 1:* The Upstream I/O Address and Upstream I/O Data registers are owned by a master.
9	Upstream I/O Control	R/W	Enables the 21554 to perform upstream indirect I/O transactions. *When 0:* The 21554 will not initiate an I/O transaction on the primary interface when the Upstream I/O Data register is accessed. The Upstream I/O Data register is treated as a reserved register. *When 1:* The 21554 is enabled to perform upstream I/O transactions when the Upstream I/O Data register is accessed with an I/O transaction. *Reset value:* 0
15:10	Reserved	R	Reserved. Read only as 0.

The following two registers provide a method for the lookup table to be accessed using I/O transactions, although memory transactions can use either this mechanism or direct access of the lookup table.

8.2.1.6 Lookup Table Offset Register

Byte Offset: 028h

Table 8-5. Lookup Table Offset Register

Bit	Name	R/W	Description
7:0	LUT_OFFSET	R/W	This register contains the byte offset of the lookup Table entry to be accessed. The access is initiated when the Lookup Table Data register is either read or written. This register should be written before the Lookup Table Data register is accessed.

8.2.1.7 Lookup Table Data Register

Byte Offset: 02F:02Ch

The lookup table is not byte-writeable; byte enables are ignored.

Table 8-6. Lookup Table Data Register

Bit	Name	R/W	Description
31:0	LUT_DATA	R/W	This register contains the data written to or read from the Lookup Table at the offset given in the Lookup Table Offset register. When this register is written, the value written is written to the specified Lookup Table entry. When this register is read, the value returned reflects the data read from the specified Lookup Table entry. The following fields are defined: • bit 0: valid bit • bits 2:1: reserved (read only as 0) • bit 3: prefetchable • bits 7:4: reserved (read only as 0) • bits 17:8: translated base or reserved (based on page size) • bits 31:18: translated base The lookup table is not reset and therefore powers up to an indeterminate value.

8.2.2 I_2O Registers

This section contains a description of the I_2O registers.

8.2.2.1 I_2O Inbound Queue

Byte Offset: 43:40h

Table 8-7. I_2O Inbound Queue

Bit	Name	R/W	Description
31:0	I_2O_IN (P) Reserved (S)	R/(WP)	This register controls the host processor access to the I_2O inbound queue. When this register is read from the primary bus, the 21554 returns the value from the head of the I_2O inbound Free_List. When this register is written from the primary bus, the 21554 writes the data to the tail of the inbound Post_List. Accesses from the secondary bus are treated as reserved. The actual location of the inbound queue lists are in local memory, and the initial location of the Free_List head and Post_List tail pointers must be programmed by the local processor in the I_2O Inbound Free_List Head Pointer and I_2O Inbound Post_List Tail Pointer registers.

8.2.2.2 I_2O Outbound Queue

Byte Offset: 47:44h

Table 8-8. I_2O Outbound Queue

Bit	Name	R/W	Description
31:0	I_2O_OUT (P) Reserved (S)	R/(WP)	This register controls the host processor access to the I_2O outbound queue. When this register is read from the primary bus, the 21554 returns the value from the head of the I_2O outbound Post_List. When this register is written from the primary bus, the 21554 writes the data to the tail of the outbound Free_List. Accesses from the secondary bus are treated as reserved. The actual location of the outbound queue lists are in local memory, and the initial location of the Post_List head and Free_List tail pointers must be programmed by the local processor in the I_2O Outbound Post_List Head Pointer and I_2O Outbound Free_List Tail Pointer registers.

8.2.2.3 I_2O Outbound Post_List Status

Byte Offset: 33:30h

Table 8-9. I_2O Outbound Post_List Status

Bit	Name	R/W	Description
2:0	Reserved	R	Reserved. Read only as 0.
3	Outbound Post Status	R	Reflects the status of the Outbound Post_List. *When 0:* The Outbound Post_List is empty. The 21554 deasserts p_inta_l (unless it is asserted for other reasons). *When 1:* The Outbound Post_List is not empty. If the Outbound Post_List Interrupt Mask bit is zero, then the 21554 asserts p_inta_l as long as this status bit is set. *Reset value:* 0
31:4	Reserved	R	Reserved. Read only as 0.

8.2.2.4 I_2O Outbound Post_List Interrupt Mask

Byte Offset: 37:34h

Table 8-10. I_2O Outbound Post_List Interrupt Mask

Bit	Name	R/W	Description
2:0	Reserved	R	Reserved. Read only as 0.
3	Outbound Post Mask	R/W	Interrupt mask for Outbound Post_List Status. *When 0:* The 21554 asserts p_inta_l when the Outbound Post_List Status bit is a 1. *When 1:* The 21554 does not assert p_inta_l when the Outbound Post_List Status bit is a 1. *Reset value:* 1
31:4	Reserved	R	Reserved. Read only as 0.

8.2.2.5 I_2O Inbound Post_List Status

Byte Offset: 3B:38h

Table 8-11. I_2O Inbound Post_List Status

Bit	Name	R/W	Description
2:0	Reserved	R	Reserved. Read only as 0.
3	Inbound Post Status	R	Reflects the status of the Inbound Post_List. *When 0:* The Inbound Post_List is empty. The 21554 deasserts s_inta_l (unless it is asserted for other reasons). *When 1:* The Inbound Post_ List is not empty. If the Inbound Post_List Interrupt Mask bit is zero, then the 21554 asserts s_inta_l as long as this status bit is set. *Reset value:* 0
31:4	Reserved	R	Reserved. Read only as 0.

8.2.2.6 I_2O Inbound Post_List Interrupt Mask

Byte Offset: 3F:3Ch

Table 8-12. I_2O Inbound Post_List Interrupt Mask

Bit	Name	R/W	Description
2:0	Reserved	R	Reserved. Read only as 0.
3	Inbound Post Mask	R/W	Interrupt mask for Inbound Post_List Status. *When 0:* The 21554 asserts s_inta_l when the Inbound Post_List Status bit is a 1. *When 1:* The 21554 does not assert s_inta_l when the Inbound Post_List Status bit is a 1. *Reset value:* 1
31:4	Reserved	R	Reserved. Read only as 0.

8.2.2.7 I_2O Inbound Free_List Head Pointer

Byte Offsets: 04B:048h

Table 8-13. I_2O Inbound Free_List Head Pointer

Bit	Name	R/W	Description
1:0	Reserved	R	Reserved. Returns 0 when read.
31:2	Inbound Free Head Ptr	R/W	Specifies the local memory Dword address of the Inbound Free_List Head Pointer. Increments when the I_2O Inbound Queue at offset 40h is read on the primary bus. This pointer automatically wraps when it reaches the upper boundary of the Inbound Free_List.

8.2.2.8 I_2O Inbound Post_List Tail Pointer

Byte Offsets: 04F:04Ch

Table 8-14. I_2O Inbound Post_List Tail Pointer

Bit	Name	R/W	Description
1:0	Reserved	R	Reserved. Returns 0 when read.
31:2	Inbound Post Tail Ptr	R/W	Specifies the local memory Dword address of the Inbound Post_List Tail Pointer. Increments when the I_2O Inbound Queue at offset 40h is written on the primary bus. This pointer automatically wraps when it reaches the upper boundary of the Inbound Post_List.

8.2.2.9 I_2O Outbound Free_List Tail Pointer

Byte Offsets: 053:050h

Table 8-15. I_2O Outbound Free_List Tail Pointer

Bit	Name	R/W	Description
1:0	Reserved	R	Reserved. Returns 0 when read.
31:2	Outbound Free Tail Ptr	R/W	Specifies the local memory Dword address of the Outbound Free_List Tail Pointer. Increments when the I_2O Outbound Queue at offset 44h is written on the primary bus. This pointer automatically wraps when it reaches the upper boundary of the Outbound Free_List.

8.2.2.10 I_2O Outbound Post_List Head Pointer

Byte Offsets: 057:054h

Table 8-16. I_2O Outbound Post_List Head Pointer

Bit	Name	R/W	Description
1:0	Reserved	R	Reserved. Returns 0 when read.
31:2	Outbound Post Head Ptr	R/W	Specifies the local memory Dword address of the Outbound Post_List Head Pointer. Increments when the I_2O Outbound Queue at offset 44h is read on the primary bus. This pointer automatically wraps when it reaches the upper boundary of the Outbound Post_List.

8.2.2.11 I_2O Inbound Post_List Counter

Byte Offsets: 05B:058h

Table 8-17. I_2O Inbound Post_List Counter

Bit	Name	R/W	Description
15:0	Inbound Post Ctr	R/(WS)	When read, returns the number of entries in the Inbound Post_List. Decrements by 1 when this location is written from the secondary interface if bit [31] of this register is written with a 0 during the same write. If bit [31] is written with a 1, then the 21554 loads the counter with the value written. Increments when the Inbound Queue at offset 40h is written from the primary interface. Reset value : 0
30:16	Reserved	R	Reserved. Read only as 0.
31	LD_IPC	W1TL(S)	Load Inbound Post_List Counter. When written with a 1 at the same time as the Inbound Post Ctr bits [15:0], loads the Inbound Post_List Counter with the value on s_ad[15:0] during that same write. When written with a 0, decrements the Inbound Post_List Counter. Reads always return 0.

8.2.2.12 I_2O Inbound Free_List Counter

Byte Offsets: 05F:05Ch.

Table 8-18. I_2O Inbound Free_List Counter

Bit	Name	R/W	Description
15:0	Inbound Free Ctr	R/(WS)	When read, returns the number of entries in the Inbound Free_List. Increments by 1 when this location is written from the secondary interface if bit [31] of this register is written with a 0 during the same write. If bit [31] is written with a 1, then the 21554 loads the counter with the value written. Decrements when the Inbound Queue at offset 40h is read from the primary interface, except when the counter is zero. The 21554 does not decrement when the counter is 0. Reset value: 0
30:16	Reserved	R	Reserved. Read only as 0.
31	LD_IFC	W1TL(S)	Load Inbound Free_List Counter. When written with a 1 at the same time as the Inbound Free Ctr bits [15:0], loads the Inbound Free_List Counter with the value on s_ad[15:0] during that same write. When written with a 0, increments the Inbound Free_List Counter. Reads always return 0.

8.2.2.13 I_2O Outbound Post_List Counter

Byte Offsets: 063:060h

Table 8-19. I_2O Outbound Post_List Counter

Bit	Name	R/W	Description
15:0	Outbound Post Ctr	R/(WS)	When read, returns the number of entries in the Outbound Post_List. Increments by 1 when this location is written from the secondary interface if bit [31] of this register is written with a 0 during the same write. If bit [31] is written with a 1, then the 21554 loads the counter with the value written. Decrements when the Outbound Queue at offset 44h is read from the primary interface, except when the counter is zero. The 21554 does not decrement when the counter is 0. Reset value: 0
30:16	Reserved	R	Reserved. Read only as 0.
31	LD_OPC	W1TL(S)	Load Outbound Post_List Counter. When written with a 1 at the same time as the Outbound Post Ctr bits [15:0], loads the Outbound Post_List Counter with the value on s_ad[15:0] during that same write. When written with a 0, increments the Outbound Post_List Counter. Reads always return 0.

8.2.2.14 I_2O Outbound Free_List Counter

Byte Offsets: 067:064h

Table 8-20. I_2O Outbound Free_List Counter

Bit	Name	R/W	Description
15:0	Outbound Free Ctr	R/(WS)	When read, returns the number of entries in the Outbound Free_List. Decrements by 1 when this location is written from the secondary interface if bit [31] of this register is written with a 0 during the same write. If bit [31] is written with a 1, then the 21554 loads the counter with the value written. Increments when the Outbound Queue at offset 44h is written from the primary interface. Reset value: 0
30:16	Reserved	R	Reserved. Read only as 0.
31	LD_OPC	W1TL(S)	Load Outbound Free_List Counter. When written with a 1 at the same time as the Outbound Free Ctr bits [15:0], loads the Outbound Free_List Counter with the value on s_ad[15:0] during that same write. When written with a 0, decrements the Outbound Free_List Counter. Reads always return 0.

8.2.3 Address Registers

All of the following registers are mapped into the 21554 configuration space and described in Section 7.2.3:

Downstream Memory 0 Translated Base Register Byte Offset:	06B:068h
Downstream I/O or Memory 1 Translated Base Register Byte Offset:	06F:06Ch
Downstream Memory 2 Translated Base Register Byte Offset:	073:070h
Downstream Memory 3 Translated Base Register Byte Offset:	077:074h
Upstream I/O or Memory 0 Translated Base Register Byte Offset:	07B:078h
Upstream Memory 1 Translated Base Register Byte Offset:	07F:07Ch

8.2.3.1 Upstream Memory 2 Lookup Table

Byte Offsets: 1FF:100h

The lookup table is not byte writeable; byte enables will be ignored.

Table 8-21. Upstream Memory 2 Lookup Table

Bit	Name	R/W	Description
	M2LUT	R/W	Contains the lookup table for the Upstream Memory 2 Base Address range. Each entry in the lookup table is 4 bytes wide. The top 16 to 24 bits, depending on the page size, of each entry are used to replace the page address of upstream memory transactions falling inside the Upstream Memory 2 BAR. The bottom four bits are control bits as described in Section 5.3.3.

8.2.4 Interrupt Registers

This section contains information about interrupt registers.

8.2.4.1 Primary Clear IRQ and Secondary Clear IRQ Registers

	Primary Clear IRQ	Secondary Clear IRQ
Byte Offset:	099:098h	09B:09Ah

These registers affect primary and secondary interrupts in the same way and are described together.

Table 8-22. Primary Clear IRQ and Secondary Clear IRQ Registers

Bit	Name	R/W	Description
15:0	CLR_IRQ	R/W1TC	This register controls the state of the Primary or Secondary Doorbell Interrupt Request bits. *When 0:* Does not cause the corresponding primary or secondary interrupt signal to be asserted. *When 1:* The primary or secondary interrupt signal is asserted if the corresponding IRQ Mask bit is zero. Writing a 1 to a bit in this register clears the corresponding interrupt request bit to 0. Writing a 0 to any bit in this register has no effect. Reading this register returns the current status of the interrupt request bits. *Reset value:* 0

8.2.4.2 Primary Set IRQ and Secondary Set IRQ Registers

	Primary Set IRQ	Secondary Set IRQ
Byte Offset:	09D:09Ch	09F:09Eh

These registers affect primary and secondary interrupts in the same way and are described together.

Table 8-23. Primary Set IRQ and Secondary Set IRQ Registers

Bit	Name	R/W	Description
15:0	SET_IRQ	R/W1TS	This register controls the state of the Primary or Secondary Doorbell Interrupt Request bits. *When 0:* Does not cause the corresponding primary or secondary interrupt signal to be asserted. *When 1:* The primary or secondary interrupt signal is asserted if the corresponding IRQ Mask bit is zero. Writing a 1 to a bit in this register sets the corresponding interrupt request bit to 1. Writing a 0 to any bit in this register has no effect. Reading this register returns the current status of the interrupt request bits. *Reset value:* 0

8.2.4.3 Primary Clear IRQ Mask and Secondary Clear IRQ Mask Registers

	Primary Clear IRQ Mask	Secondary Clear IRQ Mask
Byte Offset:	0A1:0A0h	0A3:0A2h

These registers affect primary and secondary interrupts in the same way and are described together.

Table 8-24. Primary Clear IRQ Mask and Secondary Clear IRQ Mask Registers

Bit	Name	R/W	Description
15:0	CLR_IRQM	R/W1TC	*When 0:* An interrupt is generated on the 21554's primary or secondary interface when the corresponding Primary or Secondary Interrupt Request bit is a 1. *When 1*: The corresponding interrupt request bit cannot generate an interrupt. Writing a 1 to a bit in this register clears the IRQ Mask bit to 0. Writing a 0 to any bit in this register has no effect. Reading this register returns the current status of the IRQ Mask bits. *Reset value:* FFFFFFFFh

8.2.4.4 Primary Set IRQ Mask and Secondary Set IRQ Mask Registers

	Primary Set IRQ Mask	Secondary Set IRQ Mask
Byte Offset:	0A5:0A4h	0A7:0A6h

These registers affect primary and secondary interrupts in the same way and are described together.

Table 8-25. Primary Set IRQ Mask and Secondary Set IRQ Mask Registers

Bit	Name	R/W	Description
15:0	SET_IRQM	R/W1TS	*When 0:* An interrupt is generated on the 21554's primary or secondary interface when the corresponding Primary or Secondary Interrupt Request bit is a 1. *When 1*: The corresponding interrupt request bit cannot generate an interrupt. Writing a 1 to a bit in this register sets the IRQ Mask bit to 1. Writing a 0 to any bit in this register has no effect. Reading this register returns the current status of the IRQ Mask bits. *Reset value:* FFFFFFFFh

8.2.4.5 Chip Status CSR

Byte Offsets: 083:082h

Table 8-26. Chip Status CSR

Bit	Name	R/W	Description
0	PM_D0	R/W1TC	Power Management Transition to D0. The 21554 sets this bit when it is transitioned from a low power D1 or D2 state to a high power D0 state. If the corresponding Chip IRQ Mask bit for this event is a 0, the 21554 also asserts s_inta_l to indicate to the subsystem that it is being brought to a higher power state. Writing a 1 clears this bit to a 0. Writing a 0 has no effect. *Reset value:* 0
1	Subsystem Event	R/W1TC	Generic subsystem event bit. The 21554 sets this bit when a deasserting (rising) edge is detected on s_pme_l. If s_pme_l is not used for power management purposes, then it may be used to signal some other subsystem event. If the Chip IRQ Mask bit for this event is a 0, the 21554 also asserts p_inta_l to indicate to the host system that this signal was deasserted. Writing a 1 clears this bit to a 0. Writing a 0 has not effect. *Reset value:* 0
15:2	Reserved	R	Reserved. Returns 0 when read.

8.2.4.6 Chip Set IRQ Mask Register

Byte Offsets: 085:084h

Table 8-27. Chip Set IRQ Mask Register

Bit	Name	R/W	Description
0	Set_D0M	R/W1TS	*When 0:* s_inta_l is asserted on the 21554's secondary interface when the corresponding chip event bit, indicating a return of power state to D0, is a 1. *When 1*: The corresponding chip event bit does not generate an interrupt. Writing a 1 to a bit in this register sets the Chip IRQ Mask bit to 1. Writing a 0 to any bit in this register has no effect. Reading this register returns the current status of the Chip IRQ Mask bits. *Reset value:* 1
1	Set_Sstat	R/W1TS	*When 0:* p_inta_l is asserted on the 21554's primary interface when the corresponding chip event bit, indicating a deasserting edge on s_pme_l, is a 1. *When 1*: The corresponding chip event bit does not generate an interrupt. Writing a 1 to a bit in this register sets the Chip IRQ Mask bit to 1. Writing a 0 to any bit in this register has no effect. Reading this register returns the current status of the Chip IRQ Mask bits. *Reset value:* 1
15:2	Reserved	R	Reserved. Returns 0 when read.

8.2.4.7 Chip Clear IRQ Mask Register

Byte Offsets: 087:086h

Table 8-28. Chip Clear IRQ Mask Register

Bit	Name	R/W	Description
0	Clr_D0M	R/W1TC	*When 0:* s_inta_l is asserted on the 21554's secondary interface when the corresponding chip event bit, indicating a return of power state to D0, is a 1. *When 1*: The corresponding chip event bit does not generate an interrupt. Writing a 1 to a bit in this register clears the Chip IRQ Mask bit to 1. Writing a 0 to any bit in this register has no effect. Reading this register returns the current status of the Chip IRQ Mask bits. *Reset value:* 1
1	Clr_Sstat	R/W1TC	*When 0:* p_inta_l is asserted on the 21554's primary interface when the corresponding chip event bit, indicating a deasserting edge on s_pme_l, is a 1. *When 1*: The corresponding chip event bit does not generate an interrupt. Writing a 1 to a bit in this register clears the Chip IRQ Mask bit to 0. Writing a 0 to any bit in this register has no effect. Reading this register returns the current status of the Chip IRQ Mask bits. *Reset value:* 1
15:2	Reserved	R	Reserved. Returns 0 when read.

8.2.4.8 Upstream Page Boundary IRQ 0 Register

Byte Offset: 08B:088h

Table 8-29. Upstream Page Boundary IRQ 0 Register

Bit	Name	R/W	Description
31:0	PAGE0_IRQ	R/W1TC	Each bit in this register corresponds to a page entry in the lower half of the Upstream Memory 2 range. Bit 0 corresponds to the first (lowest order) page, and bit 31 corresponds to the 32nd page. The 21554 sets the appropriate bit when it successfully transfers data to/from the initiator that addresses the last Dword in a page. If the Upstream Page Boundary 0 IRQ Mask bit corresponding to that page is zero, then the 21554 asserts s_inta_l. *Reset value:* 0

8.2.4.9 Upstream Page Boundary IRQ 1 Register

Byte Offset: 08F:08Ch

Table 8-30. Upstream Page Boundary IRQ 1 Register

Bit	Name	R/W	Description
31:0	PAGE1_IRQ	R/W1TC	Each bit in this register corresponds to a page entry in the upper half of the Upstream Memory 2 range. Bit 0 corresponds to the 33rd page, and bit 31 corresponds to the 64th (highest order) page. The 21554 sets the appropriate bit when it successfully transfers data to/from the initiator that addresses the last Dword in a page. If the Upstream Page Boundary 1 IRQ Mask bit corresponding to that page is zero, then the 21554 asserts s_inta_l. *Reset value:* 0

8.2.4.10 Upstream Page Boundary IRQ Mask 0 Register

Byte Offset: 093:090h

Table 8-31. Upstream Page Boundary IRQ Mask 0 Register

Bit	Name	R/W	Description
31:0	PAGE0_MASK	R/W	*When 0:* The 21554 asserts s_inta_l when the corresponding status bit in the Upstream Page Boundary IRQ 0 register is set. *When 1:* The 21554 does not assert s_inta_l when the corresponding status bit in the Upstream Page Boundary IRQ 0 register is set. *Reset value:* FFFFFFFFh

8.2.4.11 Upstream Page Boundary IRQ Mask 1 Register

Byte Offset: 097:094h

Table 8-32. Upstream Page Boundary IRQ Mask 1 Register

Bit	Name	R/W	Description
31:0	PAGE1_MASK	R/W	*When 0:* The 21554 asserts s_inta_l when the corresponding status bit in the Upstream Page Boundary IRQ 1 register is set. *When 1:* The 21554 does not assert s_inta_l when the corresponding status bit in the Upstream Page Boundary IRQ 1 register is set. *Reset value:* FFFFFFFFh

8.2.5 ROM Registers

This section contains information about ROM registers.

8.2.5.1 ROM Setup Register

Byte Offsets: 0C9:0C8h

Table 8-33. ROM Setup Register

Bit	Name	R/W	Description
1:0	Access Time	R/W	Number of p_clk cycles that pr_cs_l asserts low (in default mode) or pr_ale_l drives high (in multiple device mode) for a parallel ROM or other external device access. Possible values: • 00: 8 times p_clk cycle time • 01: 16 times p_clk cycle time • 10: 64 times p_clk cycle time • 11: 256 times p_clk cycle time *Reset value:* 00b
7:2	Reserved	R	Reserved. Reads only as 0.
15:8	Strobe Mask	R/W	Read and write strobe timing mask. This 8-bit field defines the setup time, duration, and hold time of pr_rd_l and pr_wr_l during the device select assertion. A 1 asserts the strobe, a 0 deasserts it. The LSB is the first bit in time, the MSB is the last bit. Each bit is one eighth of the access time, or for the following access time values: • 00: 1 p_clk cycle per bit • 01: 2 p_clk cycles per bit • 10: 8 p_clk cycles per bit • 11: 32 p_clk cycles per bit *Reset value:* 01111110b

8.2.5.2 ROM Data Register

Byte Offsets: 0CAh

Table 8-34. ROM Data Register

Bit	Name	R/W	Description
7:0	ROM_DATA	R/W	When the Parallel ROM Start bit is set, contains the read or write data for bits [7:0] of the parallel ROM. When the Serial ROM Start bit is set, contains the read or write data for bits [7:0] of the serial ROM.

8.2.5.3 ROM Address Register

Byte Offsets: 0CE:0CCh

Table 8-35. ROM Address Register

Bit	Name	R/W	Description
23:0	ROM_ADDR	R/W	Contains the byte address of the parallel ROM read or write access used when the Parallel ROM Start bit is set to a 1. Contains the byte address and Opcode used when the Serial ROM Start bit is set to a 1. The byte address is contained on bits [8:0]. The opcode is contained on bits [10:9]. Possible opcode value are: • 00: write all, erase all, write enable, programming disable • 01: write • 10: read • 11: erase When the opcode is 00b, address bits [8:7] further define the operation: • 00: write disable • 01: write all • 10: erase all • 11: write enable *Reset value:* 000400h (serial ROM read at address 0)

8.2.5.4 ROM Control Register

Byte Offsets: 0CFh

Table 8-36. ROM Control Registe

Bit	Name	R/W	Description
0	Serial ROM Start/Busy	R/W1TS	Starts a serial ROM read, write, or polling operation and returns the completion status of the access. When written with a 1, performs the serial ROM operation indicated by the serial ROM opcode. This bit is automatically cleared by the 21554 when the serial ROM access is complete. This bit should not be written unless both the Serial ROM Start and Parallel ROM Start bits are 0. Writing a 0 to this bit has no effect. If the previous serial ROM operation was a write all, erase all, write, or erase, then writing this bit causes the 21554 to poll the serial ROM to test for the completion of the operation. The result of the poll operation is reflected in bit 3 of this register. *Reset value:* 0
1	Parallel ROM Start/Busy	R/W1TS	Starts a parallel ROM read or write operation and returns the completion status of the access. When written with a 1, the 21554 performs the parallel ROM operation indicated by the Parallel ROM Read/Write Control bit. This bit is automatically set when the 21554 performs a parallel ROM read from the Primary Expansion ROM address space. This bit is automatically cleared by the 21554 when the parallel ROM access is complete. This bit should not be set unless both the Serial ROM Start and Parallel ROM Start bits are 0. Writing a 0 to this bit has no effect. *Reset value:* 0
2	Read/Write Control	R/W	Parallel ROM read/write control bit. This bit may be written with the same CSR access that sets the Parallel ROM Start bit. *When 0:* The 21554 performs a read of the parallel ROM when the Parallel ROM Start bit is set to a 1. *When 1:* The 21554 performs a write of the parallel ROM when the Parallel ROM Start bit is set to a 1. *Reset value:* 0
3	SROM_POLL	R	This bit reflects the status of the serial ROM as a result of a polling operation following a write all, erase all, write, or erase operation. This bit is set automatically by the 21554 when one of these operations is initiated, and cleared when a subsequent poll of the serial ROM indicates that the operation is complete. *Reset value:* 0
7:4	Reserved	R	Reserved. Reads only as 0.

8.2.6 Control and Miscellaneous Registers

This section contains information about control and miscellaneous registers.

8.2.6.1 Scratchpad 0 Through Scratchpad 7 Registers

Scratchpad 0 Byte Offset:	0AB:0A8h
Scratchpad 1 Byte Offset:	0AF:0ACh
Scratchpad 2 Byte Offset:	0B3:0B0h
Scratchpad 3 Byte Offset:	0B7:0B4h
Scratchpad 4 Byte Offset:	0BB:0B8h
Scratchpad 5 Byte Offset:	0BF:0BCh
Scratchpad 6 Byte Offset:	0C3:0C0h
Scratchpad 7 Byte Offset:	0C7:0C4h

These registers may be used as read/write scratchpad registers.

Table 8-37. Scratchpad 0 Through Scratchpad 7 Registers

Bit	Name	R/W	Description
31:0	SCRATCH0	R/W	32-bit scratchpad register 0.
31:0	SCRATCH1	R/W	32-bit scratchpad register 1.
31:0	SCRATCH2	R/W	32-bit scratchpad register 2.
31:0	SCRATCH3	R/W	32-bit scratchpad register 3.
31:0	SCRATCH4	R/W	32-bit scratchpad register 4.
31:0	SCRATCH5	R/W	32-bit scratchpad register 5.
31:0	SCRATCH6	R/W	32-bit scratchpad register 6.
31:0	SCRATCH7	R/W	32-bit scratchpad register 7.

I_2O Support 9

The 21554 implements an I_2O messaging unit to allow passing of I_2O messages between the host system and the local subsystem (called IOP in I_2O nomenclature).

For the host system to identify the local subsystem as an I_2O compliant IOP, the class code must be preloaded to indicate I_2O support. The Base Class is loaded with the code for intelligent I/O controllers (0Eh). The Sub-class Code is loaded with the code indicating I_2O conformance (00h). The Programming Interface is loaded with 01h to indicate 32-bit data width, 32-bit addressing, little endian, with support of the outbound post status and mask registers. The I_2O Enable bit in the Chip Control 1 configuration register must be set to a 1 to enable the I_2O message unit. Otherwise, accesses to the I_2O Inbound Queue and I_2O Outbound Queue result in TRDY# and a discard of data for memory writes, and TRDY# and return of FFFFFFFFh for memory reads.

The 21554 implements two predefined I_2O registers in CSR space that allow access to the I_2O Inbound Queue (at offset 40h) and the I_2O Outbound Queue (at offset 44h). The actual queues are located in local memory. Each queue has a Post_List FIFO and a Free_List FIFO. The Post_List contains I_2O message frame addresses (MFAs). The Free_List contains empty MFAs. The 21554 implements hardware to control the FIFOs from the host side. Control of the FIFOs from the local processor side is done in software.

9.1 Inbound Message Passing

An inbound message is passed from the host processor to the local processor in the following steps:

1. The host processor removes an empty MFA, if available, from the head of the Inbound Free_List.
2. The host processor posts an MFA containing the address of the message frame to the tail of the Inbound Post_List.
3. The I_2O controller interrupts the local processor, indicating that an MFA exists in the Inbound Post_List.
4. The local processor retrieves the MFA from the head of the Inbound Post_List.
5. After the local processor consumes the message, it replaces the empty MFA onto the tail of the Inbound Free_List.

The 21554 implements the following hardware for the Inbound Queue:

- Inbound Queue register at CSR offset 40h
- Inbound Free_List head pointer at CSR offset 48h
- Inbound Post_List tail pointer at CSR offset 4Ch
- Inbound Post_List counter at CSR offset 58h
- Inbound Free_List counter at CSR offset 5Ch
- Inbound Post_List status at CSR offset 38h
- Inbound Post_List interrupt mask at CSR offset 3Ch

When the host processor has a message to pass to the local processor, it first reads the Inbound Queue location at 40h to remove an empty MFA from the Inbound Free_List. The 21554 maintains an on silicon 2-Dword buffer to hold the next two empty MFAs from the Inbound Free_List. If this buffer is not empty, the 21554 returns TRDY# and the next empty MFA from its buffer. If the internal buffer empties as a result of this read operation, then the 21554 automatically reads one or two more MFAs from the Inbound Free_List if the Inbound Free_List counter is non-zero. The 21554 decrements the Inbound Free_List counter by either 2 or 1, respectively, when the read completes on the secondary bus. The 21554 increments the Inbound Free_List head pointer by either 1 or 2 Dwords, respectively, as well.

If the Inbound Free_List counter is zero and the prefetch buffer is empty, then the 21554 immediately returns FFFFFFFFh to the host and does not decrement the Inbound Free_List counter or increment the Inbound Free_List head pointer.

Once the host obtains an empty MFA from the Inbound Free_List, it may then post a message to the local processor. The host processor posts a message to the Inbound Queue by writing the MFA to offset 40h. The 21554 treats the write to location 40h as a posted write; that is, it returns TRDY# to the initiator and places the write data in the posted write queue. The 21554 translates the address to the current value of the Inbound Post_List tail pointer. Once the MFA is queued in the posted write buffer, the 21554 increments the Inbound Post_List tail pointer by 1. The 21554 can continue to accept posted inbound MFAs as long as there is room in the downstream posted write queue. The 21554 performs a secondary bus memory write of this data to the location addressed by the Inbound Post_List tail pointer. When this write is completed on the secondary bus, the 21554 increments the Inbound Post_List counter by 1. As long as the Inbound Post_List counter is non-zero, the 21554 sets the Inbound Post_List status to 1, and asserts s_inta_l if the Inbound Post_List Mask bit is zero. s_inta_l remains asserted until either the Inbound Post_List counter is zero, or the Inbound Post_List Mask bit is set.

The local processor manages the removal of the MFA from the Inbound Post_List and the replacement of the empty MFA to the Inbound Free_List in software. The 21554 does not implement inbound queue pointers that would be used by the local processor. However, the local processor must manage the 21554's Inbound Post_List counter when it removes an MFA. When the local processor removes the message from the Inbound Post_List, it must decrement the Inbound Post_List counter by writing bit [31] of that register with a 0. If the counter decrements to zero, the 21554 deasserts s_inta_l, indicating that there are no more posted MFAs in the Inbound Queue.

Once the local processor consumes the inbound message from the host, it replaces the empty MFA onto the end of the Inbound Free_List. Again, the 21554 does not manage the pointers for this operation, so the local processor must manage this in software. However, the local processor manages the 21554's Inbound Free_List counter when it replaces an empty MFA. When the local processor replaces the empty MFA to the Inbound Free_List, it must write bit [31] of the Inbound Free_List counter with a zero, which causes the 21554 to increment the counter by 1.

9.2 Outbound Message Passing

An outbound message is passed from the local processor to the host processor in the following steps:

1. The local processor removes an empty MFA, if available, from the head of the Outbound Free_List.
2. The local processor posts an MFA containing the address of the message frame to the tail of the Outbound Post_List.
3. The I_2O Controller interrupts the host processor, indicating that an MFA exists in the Outbound Post_List.
4. The host processor retrieves the MFA from the head of the Outbound Post_List.
5. After the host processor consumes the message, it replaces the empty MFA onto the tail of the Outbound Free_List.

The 21554 implements the following hardware for the Outbound Queue:

- Outbound Queue register at CSR offset 44h
- Outbound Free_List tail pointer at CSR offset 50h
- Outbound Post_List head pointer at CSR offset 54h
- Outbound Post_List counter at CSR offset 60h
- Outbound Free_List counter at CSR offset 64h
- Outbound Post_List status at CSR offset 30h
- Outbound Post_List interrupt mask at CSR offset 34h

When the local processor has a message to pass to the host processor, it first must remove an empty MFA from the head of the Outbound Free_List. The 21554 does not implement outbound queue pointers that are used by the local processor. However, the local processor manages the 21554's Outbound Free_List counter when it removes an empty MFA. When the local processor removes the empty MFA from the Outbound Free_List, it must write bit [31] of the Outbound Free_List counter to a zero, which causes the 21554 to decrement the Outbound Free_List counter by 1.

When the local processor posts a message to the Outbound Post_List, it must write bit [31] of the Outbound Post_List counter to a zero, which causes the 21554 to increment the counter by 1. A non-zero value in the Outbound Post_List counter indicates that the Outbound Post_List contains MFAs intended for the host processor. Upon detection of a non-zero value in the Outbound Post_List counter, the 21554 sets the Outbound Post_List status to 1 and asserts p_inta_l (if the Outbound Post_List Mask bit is zero), to indicate to the host processor that one or more MFAs exist in the Outbound Post_List. Signal p_inta_l remains asserted until either the Outbound Post_List counter is zero and the outbound prefetch buffer empties, or the Outbound Post_List Mask bit is set.

The host processor removes the message from the Outbound Post_List by reading the 21554 CSR offset 44h. The 21554 maintains a 2-Dword outbound prefetch buffer to hold the next two MFAs from the Outbound Post_List. If this buffer is not empty, the 21554 returns TRDY# and the next MFA from its buffer. If the internal buffer empties as a result of this read operation, then the 21554 automatically reads one or two more MFAs from the Outbound Post_List if the counter is non-zero. When the read completes on the secondary bus, the 21554 decrements the Outbound Post_List counter by 1 or 2, respectively. The 21554 increments the Outbound Post_List head pointer by either 1 or 2 Dwords, respectively, as well.

If the Outbound Post_List counter is zero and the prefetch buffer is empty, the 21554 immediately returns FFFFFFFFh to the host and does not decrement the Outbound Post_List counter. If the counter decrements to zero and the 2-Dword prefetch buffer is empty, the 21554 deasserts p_inta_l, indicating that there are no more posted MFAs in the Outbound Queue.

Once the host processor consumes the outbound message from the local processor, it replaces the empty MFA onto the end of the Outbound Free_List. When the host processor replaces the empty MFA to the Outbound Free_List, it writes the Outbound Queue at the 21554 CSR offset 44h. The 21554 treats the write to location 44h as a posted write; that is, it returns TRDY# to the initiator and places the write data in the downstream posted write queue. The 21554 translates the address to the current value of the Outbound Free_List tail pointer. The 21554 increments the Outbound Free_List tail pointer. The 21554 can continue to accept write to this address as long as there is room in the downstream posted write queue. The 21554 writes the data to the secondary bus location addressed by the Outbound Free_List tail pointer. When the write is completed, the 21554 increments the Outbound Free_List counter.

9.3 Miscellaneous Notes

Read transactions to I_2O Inbound and Outbound Queues at 40h and 44h are not ordered with respect to transactions in the posted write or delayed transaction queues. Reads to these two registers will not flush posted writes. Write transactions to these registers are placed in the posted write queue, and follow the same ordering rules as other posted memory writes.

If the 21554 detects parity errors, master abort, or target abort during a read or write access to the Inbound or Outbound Queues, the 21554 treats the error condition the same way as it does any other delayed read or posted write.

The 21554 queue pointers support FIFO sizes of 256, 512, 1K, 2K, 4K, 8K, 16K, and 32K entries. The number of entries is selectable in the Chip Control 1 Configuration Register. The wrap function for all of the I_2O pointers maintained by the 21554 is performed in hardware, therefore all FIFOs must be located in an aligned address boundary. All MFA counters maintained by the 21554 may be individually loaded with any data value by writing a 1 to bit 31 of the corresponding counter Dword offset. When either the Inbound Free_List counter or the Outbound Post_List counter is loaded, the 21554 will also discard any prefetched data in the corresponding prefetch buffer. Note that the 21554 actions are unpredictable in response to primary bus transactions addressing the I_2O Outbound or Inbound Queues while a corresponding counter load is occurring on the secondary bus.

The 21554 I_2O counters are consistent with the number of entries in the various lists from the secondary bus, or local memory, point of view. This means that counters do not include any MFAs that exist in the 21554 posting or prefetch buffers. The counters include only those entries that are present in local memory.

Interrupt and Scratchpad Registers 10

10.1 Interrupt Support

The 21554 supports hardware to facilitate software-generated interrupts, as well as interrupts initiated by the 21554 activity. The 21554 has interrupt request status and interrupt mask bits for the following conditions:

- I_2O Inbound Post_List FIFO not empty.
  - — Cleared automatically when FIFO is empty
  - — Asserts s_inta_l when the corresponding mask bit is zero
- I_2O Outbound Post_List FIFO not empty.
  - — Cleared automatically when FIFO is empty
  - — Asserts p_inta_l when the corresponding mask bit is zero
- Upstream memory read or write Dword transfer in Upstream Memory Range 2 addresses the last Dword in a page.
  - — One event and enable bit for each of the 64 pages in Upstream Memory Range 2
  - — Cleared by writing a 1 to the corresponding status bit in the Upstream Page Boundary IRQ 0 or 1 registers
  - — Asserts s_inta_l when the corresponding mask bit is zero
- A subsystem event is indicated by a rising edge on s_pme_l.
  - — Cleared by writing a 1 to the corresponding status bit in the Chip Status CSR
  - — Asserts p_inta_l when the corresponding mask bit is zero
- A power management transition from state D1 or D2 to state D0 occurs.
  - — Cleared by writing a 1 to the corresponding status bit in the Chip Status CSR
  - — Asserts s_inta_l when the corresponding mask bit is zero

10.2 Doorbell Interrupts

A 16-bit software controlled interrupt request register and an associated 16-bit mask register is implemented for each interface (primary and secondary). Each register is byte addressable so they can be used as two sets of 8-bit interrupt request and interrupt mask registers for each interface (four in all) if desired. These registers can be accessed from the primary or secondary interface of the 21554, in either memory space or I/O space.

The 21554 doorbell interrupt functionality consists of the following registers:

- Primary Interrupt Request 16-bit register
- Secondary Interrupt Request 16-bit register
- Primary Interrupt Request (IRQ) Mask 16-bit register
- Secondary Interrupt Request (IRQ) Mask 16-bit register

The primary interrupt pin, p_inta_l, is asserted low whenever one or more Primary Interrupt Request bits are set and their corresponding Primary IRQ Mask bits are 0. p_inta_l remains asserted as long as this condition exists. p_inta_l is deasserted when either the Primary Interrupt Request bit is cleared or the Primary IRQ Mask bit is set. The secondary interrupt pin, s_inta_l, is asserted low whenever one or more Secondary Interrupt Request bits are set and their corresponding Secondary IRQ Mask bits are 0, and remains asserted as long as this condition exists. s_inta_l is deasserted when either the Secondary Interrupt Request bit is cleared or the Secondary IRQ Mask bit is set.

Each register can be accessed at two addresses. One location is used to set bits and the other location is used to clear them. To modify a request bit, a 1 is written to the bit in either the write-1-to-set interrupt or write-1-to-clear interrupt register address. Interrupt status can be read from either register.

10.3 Scratchpad Registers

The 21554 implements eight 32-bit scratchpad registers that can be accessed in either memory or I/O space from either the primary or secondary interface. These registers can be used to pass control and status information between primary and secondary bus devices, or treated as generic R/W registers. Writing or reading a scratchpad register does not cause an interrupt to be asserted. Doorbell interrupts can be used for this purpose.

Parallel ROM Interface 11

The 21554 supports the attachment of a standard parallel ROM or EPROM with the addition of a small amount of external logic. Flash ROMs compatible with Intel's 28F00x can be used with this interface. The 21554 supports a PCI expansion ROM Base Address Register on its primary interface with ROM sizes of 4KB to 16MB. Using these features the 21554 can provide the PCI expansion ROM interface for the subsystem. If the local subsystem does not require a PCI expansion ROM, the expansion ROM BAR can be disabled.

Once the host completes the configuration of the primary PCI interface, the parallel ROM may be accessed by the host in the memory address range assigned by the PCI expansion ROM Base Address Register. The parallel ROM is not directly accessible in the memory address space of the secondary PCI interface. However, the parallel ROM can be indirectly accessed from both the primary and secondary PCI interfaces through the 21554 CSRs.

Note: The parallel ROM cannot support simultaneous access using the Primary expansion ROM BAR and the ROM CSR registers at the same time. Results will be unpredictable. Additionally, the ROM should not be accessed simultaneously from the primary and secondary interface using the ROM CSR registers; otherwise results will be unpredictable.

The 21554 expansion ROM interface signals are listed in Table 11-1. The ROM address is driven out on the 8-bit data bus in three consecutive cycles. External octal D registers with active low enables are required to capture the ROM address.

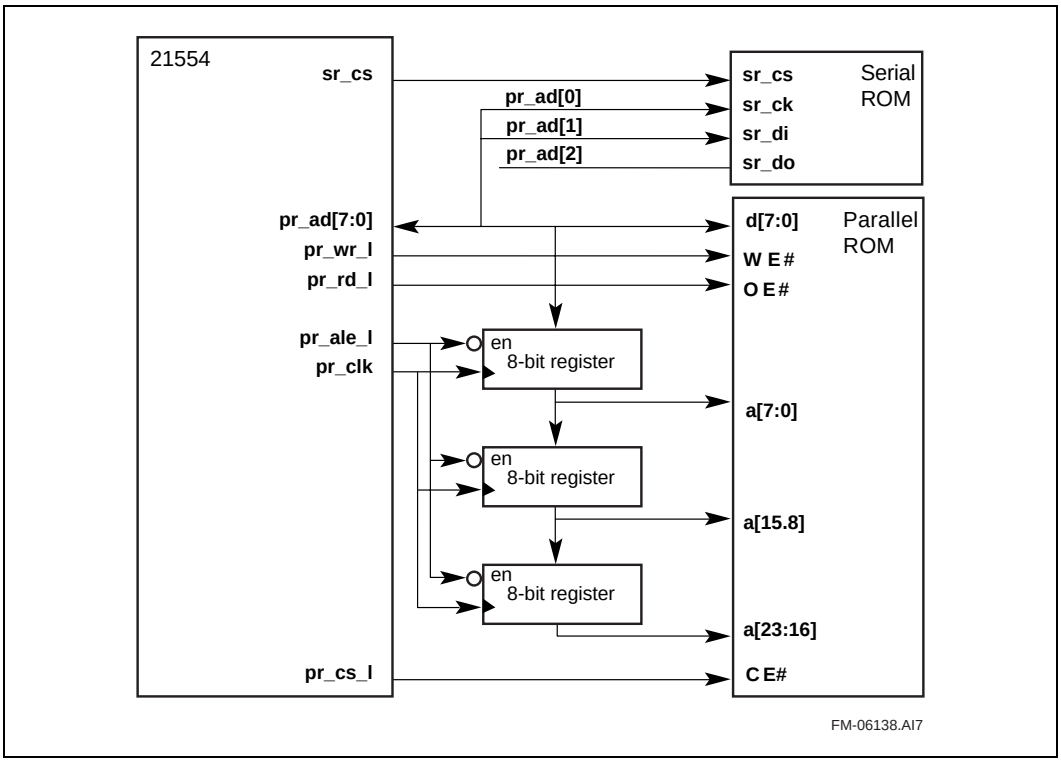
The serial ROM data and clock signals are multiplexed with the ROM signals. A description of the serial ROM interface is given in Chapter 12. The parallel ROM can also be used to interface other slave-only devices to the ROM address and data bus. This configuration is described in Section 11.5.

Table 11-1. Parallel ROM Interface Signals

Name	Type	Description
pr_ad[7:0]	TS	Multiplexed pins containing • Address lines [23:16] during the first address cycle • Address lines [15:8] during the second address cycle • Address lines [7:0] during the third address cycle • Data lines [7:0] during the data cycle
pr_rd_l	O	ROM output enable
pr_wr_l	O	ROM write enable
pr_cs_l	O	ROM chip select
pr_ale_l	O	Address register clock enable
pr_clk	O	Address register clock, p_clk divided by 2

Figure 11-1 shows how a parallel and serial ROM can be connected to the 21554. This figure illustrates the connection of a 16MB ROM. If a smaller ROM is used, the address registers corresponding to the upper address bits can be eliminated, as those upper address bits are ignored.

Figure 11-1. Parallel and Serial ROM Connections



11.1 Parallel ROM Read by CSR Access

Byte reads of the parallel ROM may be performed by CSR access of the ROM Control register, ROM Address register, and ROM Data register. A byte read is performed as follows:

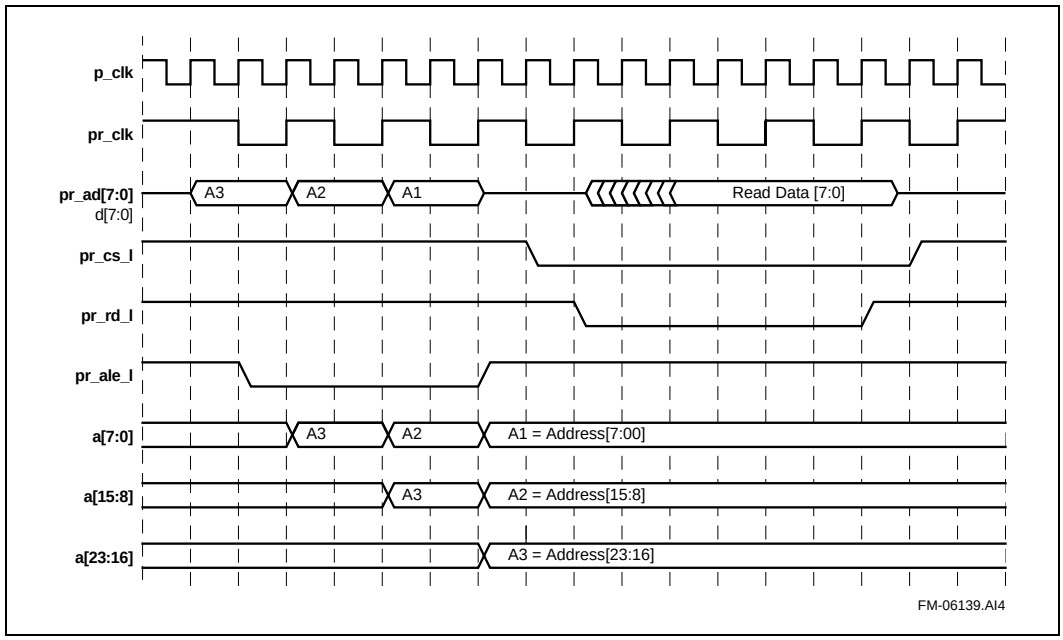
1. The initiator writes the byte address offset to the ROM Address register.
2. The initiator writes the Parallel ROM Start bit to a 1, Serial ROM Start bit to a 0, and the ROM Read/Write Control bit to a 0 in the ROM Control register.
3. When the initiator reads the Parallel ROM Start bit in the ROM Control register as a 0, the ROM operation is complete and the initiator can obtain the read data from the ROM Data register.

When a byte read of the parallel ROM is performed, the 21554 follows this sequence on the ROM interface, also shown in Figure 11-2.

1. The 21554 drives address bits [23:16] on the pr_ad[7:0] pins and asserts pr_ale_l to enable the address registers.
2. The 21554 drives pr_clk high, latching address bits [23:16] into the first external register.
3. The 21554 drives pr_clk low.
4. The 21554 drives address bits [15:8] on the pr_ad[7:0] pins.

5. The 21554 drives pr_clk high, latching address bits [15:8] into the first external register, and address bits [23:16] into the second external register.
6. The 21554 drives pr_clk low.
7. The 21554 drives address bits [7:0] on the pr_ad[7:0] pins.
8. The 21554 drives pr_clk high, latching address bits [7:0] into the first external register, address bits [15:8] into the second external register, and address bits [23:16] into the third external register. All the ROM address bits are now driven to the appropriate ROM pins.
9. The 21554 deasserts the address register enable, pr_ale_l.
10. The 21554 asserts the pr_cs_l and pr_rd_l pins according to the strobe setup timing specified by the Strobe Mask in the ROM Setup register.
11. The parallel ROM drives read data onto the pr_ad[7:0] pins.
12. The 21554 samples the read data and deasserts pr_rd_l as specified by the strobe mask. The 21554 also deasserts pr_cs_l according to the access time specified in the ROM Setup register.
13. The 21554 clears the Parallel ROM Start bit in the ROM Control register.
14. Valid data can now be read from the ROM Data register.

Figure 11-2. Parallel ROM Read Timing



11.2 Parallel ROM Write by CSR Access

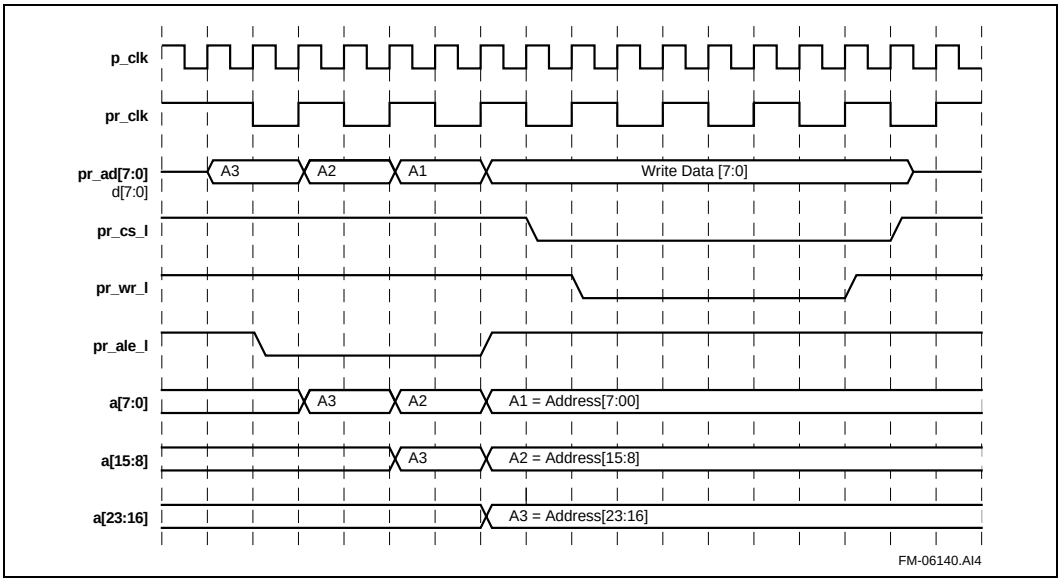
Byte writes of the parallel ROM may be performed by CSR access of the ROM Control register, Parallel ROM Address register, and ROM Data register. A byte write is performed as follows:

1. The initiator writes the byte address offset to the ROM Address register.
2. The initiator writes one byte of write data into the ROM Data register.
3. The initiator writes the Parallel ROM Start bit to a 1, Serial ROM Start bit to a 0, and the ROM Read/Write Control bit to a 1 in the ROM Control register. This may be done with the same CSR access.
4. When the initiator reads the Parallel ROM Start bit in the ROM Control register as a 0, the access is complete.

When a byte write to the parallel ROM is performed, the 21554 follows this sequence on the ROM interface, also shown in Figure 11-3.

1. The 21554 drives address bits [23:16] on the pr_ad[7:0] pins and asserts the address register enable, pr_ale_l.
2. The 21554 drives pr_clk high, latching address bits [23:16] into the first external register.
3. The 21554 drives pr_clk low.
4. The 21554 drives address bits [15:8] on the pr_ad[7:0] pins.
5. The 21554 drives pr_clk low, latching address bits [15:8] into the first external register, and address bits [23:16] into the second external register.
6. The 21554 drives pr_clk low.
7. The 21554 drives address bits [7:0] on the pr_ad[7:0] pins.
8. The 21554 drives pr_clk high, latching address bits [7:0] into the first external register, address bits [15:8] into the second external register, and address bits [23:16] into the third external register. All the ROM address and control bits are now driven to the appropriate ROM pins.
9. The 21554 deasserts the address register enable, pr_ale_l.
10. The 21554 drives the write data on pr_ad[7:0].
11. The 21554 asserts the pr_cs_l and pr_wr_l pins according to the strobe setup timing specified by the Strobe Mask in the ROM Setup register.
12. The 21554 deasserts pr_wr_l according to the strobe timing in the ROM Setup register, and deasserts the pr_cs_l according to the access time in the ROM Setup register.
13. The 21554 clears the Parallel ROM Start bit in the ROM Control register.

Figure 11-3. Parallel ROM Write Timing



11.3 Parallel ROM Dword Read

A Dword read is performed on the parallel ROM interface when a read is initiated on the primary bus whose address falls into the address range defined by the Primary expansion ROM Base Address register. The 21554 treats a memory read through the Primary expansion ROM BAR as a delayed read and returns a target retry to the initiator. The 21554 performs four consecutive byte reads of the ROM. Once the four byte reads are complete, then the 21554 returns the read data to the initiator on the next read attempt to that address to complete the delayed transaction. The 21554 automatically sets the Parallel ROM Start/Busy bit upon initiation of the read and clears the bit when the ROM read is complete.

Note: The CSR access method should not be used for the parallel ROM when reads to the parallel ROM through the Primary expansion ROM BAR are taking place.

11.4 Access Time and Strobe Control

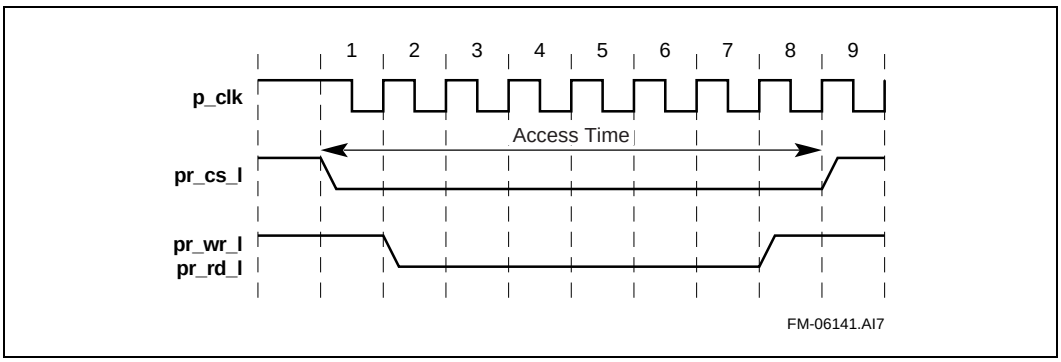
The 21554 controls both the access time and the read and write strobe timing through the ROM Setup CSR.

The access time is specified as a multiple of the p_clk signal, and must be set to 8, 16, 64, or 256 times the length of a p_clk cycle. This specifies the number of p_clk cycles that the 21554 asserts pr_cs_l. The reset value is 8 times the p_clk cycle time.

The read and write strobe timing of pr_rd_l and pr_wr_l is given as an 8-bit mask. Each bit corresponds to one eighth of the access time, which can be 1, 2, 8, or 32 p_clk cycles. Bit 0 corresponds to the first cycle. When a bit is a 0, the read or write strobe is deasserted, When the bit is a 1, the read or write strobe is asserted. pr_cs_l is asserted at the beginning of the first cycle and deasserted at the end of the last cycle. The reset value for the strobe mask is 01111110b. Since the reset value of the access time is 8 p_clk cycles, this means a read or write access has the following timing, also shown in Figure 11-4.

- Cycle 1: assert pr_cs_l
- Cycle 2 through 7: both pr_cs_l and pr_rd_l or pr_wr_l asserted
- Cycle 8: deassert pr_rd_l or pr_wr_l
- Cycle 9: deassert pr_cs_l

Figure 11-4. Read and Write Strobe Timing



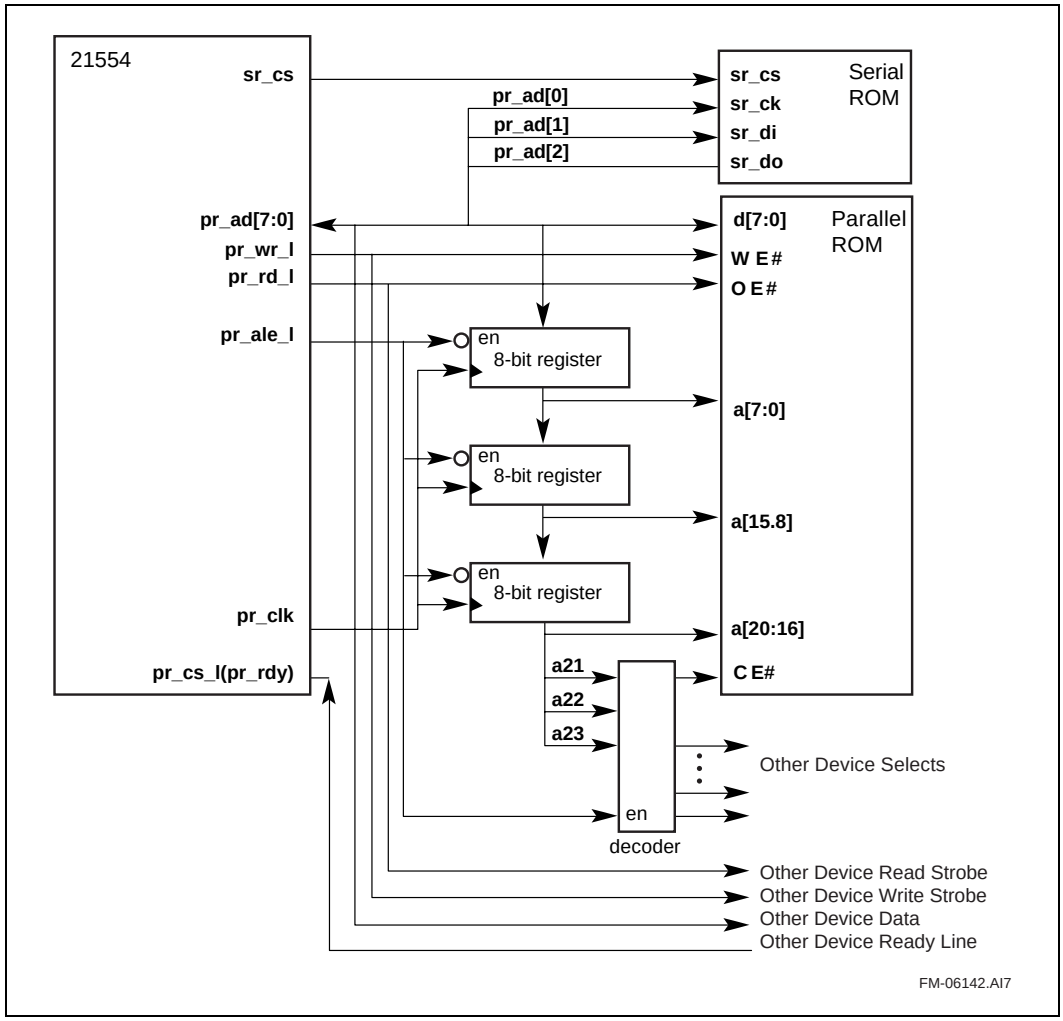
11.5 Attaching Additional Devices to the ROM Interface

The 21554 allows additional devices to be attached to the ROM interface. Two ROM interface signals are slightly redefined in order to support multiple devices by setting the Multiple Device Enable bit in the Chip Control 0 configuration register. In this mode, the maximum ROM size is reduced due to use of the upper address lines to decode device select lines.

For this mode of operation, an external decoder is needed to decode the upper address bits into device select lines. The pr_ale_l signal is used to enable the device select decoder, in addition to enabling the clocking of the address registers. When pr_ale_l is low, the address registers are enabled, and when pr_ale_l is high, the device select decoder is enabled. Figure 11-5 illustrates how the interface is connected for use with multiple devices.

In addition, in this mode the pr_cs_l output pin is redefined to be a device ready input (pr_rdy). Note that if multiple devices are attached and multiple device mode is used, pr_cs_l should be pulled up through an external resistor. When pr_rdy is deasserted, the device select signal (controlled by pr_ale_l) and read or write strobe assertions are extended until pr_rdy is asserted again. The read and write strobes are deasserted upon detection of chip select deassertion (falling pr_ale_l) with the hold time specified by the strobe mask.

Figure 11-5. Attaching Multiple Devices on the ROM Interface



Serial ROM Interface 12

The serial ROM interface is used to preload data into the 21554 configuration registers with vendor-specific values. The format for the serial ROM data is given in Section 20.1. The serial ROM may also be used to support the Vital Product Data (VPD) interface as described in Chapter 18. The serial ROM interface consists of four signals, as shown in Table 12-1. The chip select, sr_cs, has a dedicated pin. The other signals are multiplexed with the parallel ROM interface signals. The serial ROM may be attached directly to the serial ROM pins without additional external logic.

Table 12-1. Serial ROM Interface Signals

Name	Type	Description	21554 Pin
sr_cs	O	Serial ROM Chip Select	sr_cs
sr_ck	O	Serial ROM Clock	pr_ad[0]
sr_di	O	Serial ROM Data In	pr_ad[1]
sr_do	I	Serial ROM Data Out	pr_ad[2]

The serial ROM interface works with the Microchip 93LC66A* or compatible SROM, which is a byte-organized Microwire SROM with 4 Kb (512 bytes) of storage. The clock input to the serial ROM is the primary clock input, p_clk, at a maximum clock frequency of 33 MHz, divided by 34. The duty cycle is approximately 50%.

12.1 Serial ROM Preload Operation

The serial ROM interface is used to preload the 21554 configuration registers whenever the 21554 configuration registers are reset, either through assertion of p_rst_l, by setting the Chip Reset bit in the Chip Control Register, or after a power management transition from $D3_{hot}$ to D0. Once reset is complete, by detecting p_rst_l deasserted and the Chip Reset bit reset to 0, the 21554 automatically starts a serial read from the ROM. All the 21554 initialization data is loaded with a single read operation, by keeping the chip select asserted and toggling the clock. The 21554 returns a target retry to all configuration accesses until the preload operation is complete. The preload operation takes approximately 550 serial ROM clock cycles. For a detailed description of the preload sequence, see Chapter 6.

If the serial ROM is not present, the sr_do (pin pr_ad[2]) should be pulled up through an external resistor. If the serial ROM is present but register preload is not desired, bits [7:6] can be any value except the preload enable sequence 10b. If the 21554 does not detect the preload enable sequence when reading the first byte, it stops the preload operation. In this case, all configuration registers preloaded with the serial ROM remain at their reset value and should be initialized by the local processor before host access.

12.2 Serial ROM Operation by CSR Access

The 21554 allows serial ROM access through CSR control. A serial ROM operation consists of the following three phases:

1. Command phase of 3 bits
2. Address phase of 9 bits
3. Data phase of 8 or more bits
   - — Read operations may consist of any number of data bits.
   - — Write operations always consist of 8 data bits.

For a read type operation, the data is driven from the serial ROM to the 21554 on sr_do. For a write operation, the data is driven from the 21554 to the serial ROM on sr_di.

To perform a serial ROM access, the initiator should make sure that both the parallel and serial ROM Start/Busy bits are clear in the ROM Control CSR. The serial ROM word address and the serial ROM opcode are then written to the ROM Address CSR. Defined opcodes are:

00	Write enable, write disable, write all, erase all
01	Write
10	Read
11	Erase

For opcode 00, a byte address is not used, and the two most significant address bits [8:7] distinguish between the four commands:

00	Write disable
01	Write all
10	Erase all
11	Write enable

Prior to a serial ROM write or write all transaction, the 8-bit write data must be written in the ROM Data CSR.

To initiate the serial ROM access, the Serial ROM Start bit in the ROM Control CSR is written with a 1 (the parallel ROM Start bit must be written to a 0 with this access). The 21554 then initiates the serial ROM access. When the serial ROM access is complete, the 21554 automatically clears the Serial ROM Start bit. If the operation is a read, the data then can be read from the ROM Data register.

The write, write all, erase, and erase all commands may take 10 ms or more to complete internal to the ROM. A poll of the serial ROM must be performed to discover whether these operations are complete. For these commands, when the serial ROM access is initiated, the 21554 also sets the SROM_POLL bit in the ROM Control register. This bit remains asserted after the 21554's access to the serial ROM completes. The serial ROM must be polled by CSR access and return a ready indication in order to clear the SROM_POLL bit.

The serial ROM is polled by the 21554 when the Serial ROM Start bit is written with a 1 when the SROM_POLL bit is set. The 21554 asserts sr_cs and drives sr_di (pin pr_ad[1]) low. When the serial ROM drives sr_do (pin pr_ad[2]) high in response, it has completed the operation internally and the 21554 clears the SROM_POLL bit. The serial ROM is now ready for another access. Note that the SROM_POLL bit *must* be set for the 21554 to poll the serial ROM, otherwise the 21554 initiates another serial ROM access if the Serial ROM Start bit is written.

A summary of the actions needed for a serial ROM read access follows:

1. The initiator writes the byte address and the opcode in the ROM Address CSR.
2. The initiator writes the Serial ROM Start bit to a 1 and the Parallel ROM Start bit to a 0 in the ROM Control CSR.
3. When the Serial ROM Start bit in the ROM Control CSR is read as a zero, the initiator may read the 8-bit data from the ROM Data register.

A summary of the actions needed for a write operation follows:

1. The initiator writes the byte address and the opcode in the ROM Address CSR.
2. The initiator writes the 8-bit data in the ROM Data CSR.
3. The initiator writes the Serial ROM Start bit to a 1 and the Parallel ROM Start bit to a 0 in the ROM Control CSR in the same CSR access.
4. When the Serial ROM Start bit in the ROM Control CSR is read as a zero, the initiator polls the serial ROM to test for write completion by writing the Serial ROM Start bit to a 1.
5. When the Serial ROM Start bit in the ROM Control CSR is read as a zero, the SROM_POLL bit indicates the status of the polling operation. If SROM_POLL is read as a one, the serial ROM should be polled again. If SROM_POLL is read as a 0, the operation is complete.

The erase, erase all, write enable, and write disable all use the write protocol. For all these operations, however, the ROM Data register does not need to be written. In addition, the write enable and write disable operations do not require polling for completion. The following figures show the timing diagrams for serial ROM read, write, write all, write enable, write disable, erase, erase all, and check status (polling) operations.

Figure 12-1. Serial ROM Read Timing Diagram

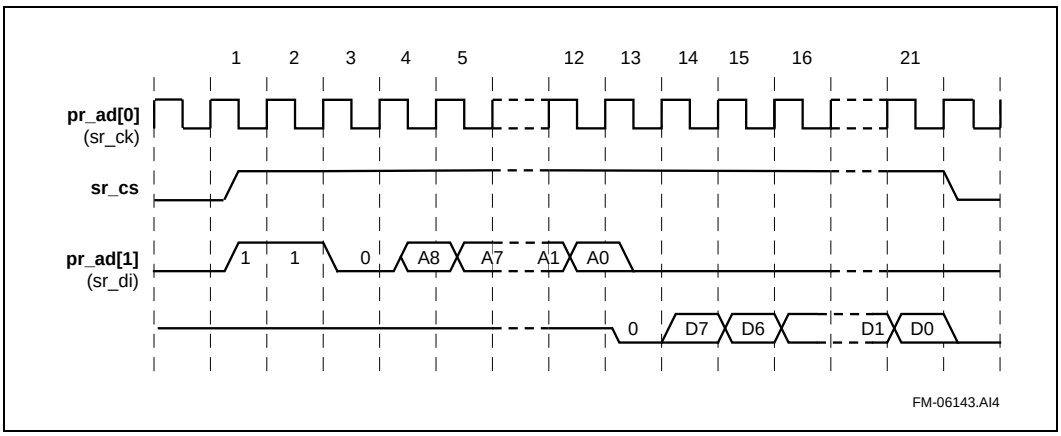


Figure 12-2. Serial ROM Write Timing Diagram

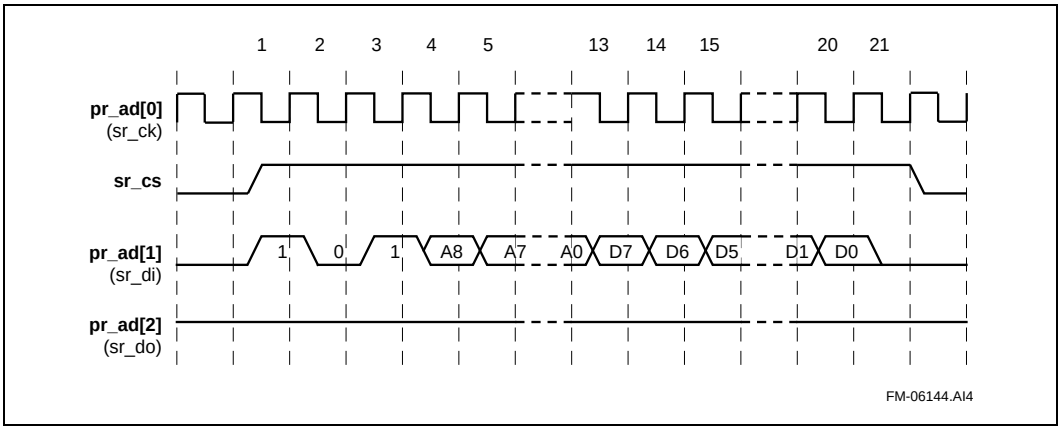


Figure 12-3. Serial ROM Write All Timing Diagram

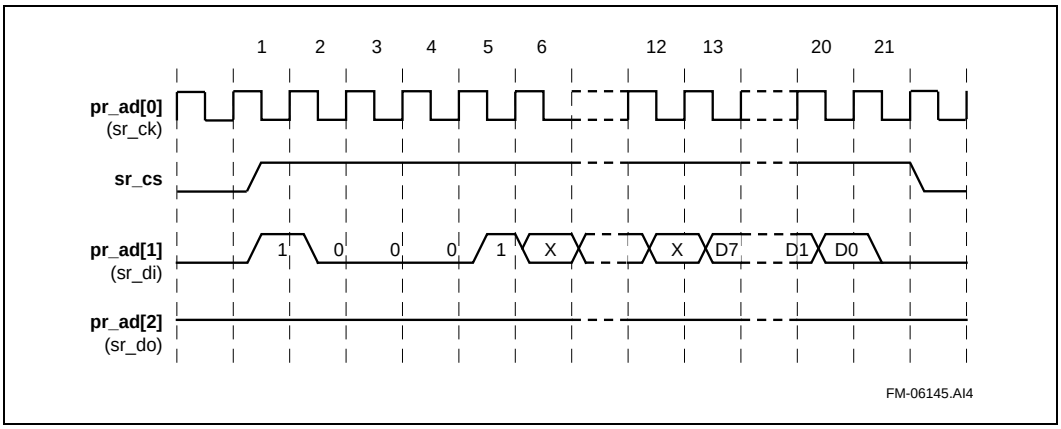


Figure 12-4. Serial ROM Write Enable Timing Diagram

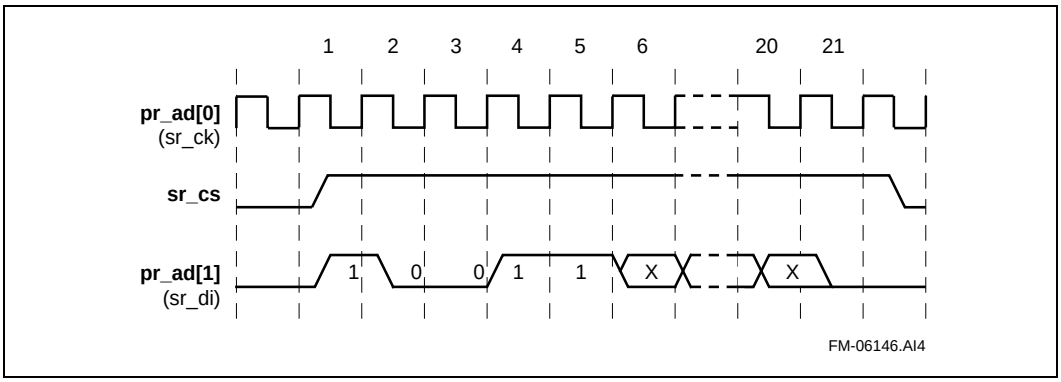


Figure 12-5. Serial ROM Write Disable Timing Diagram

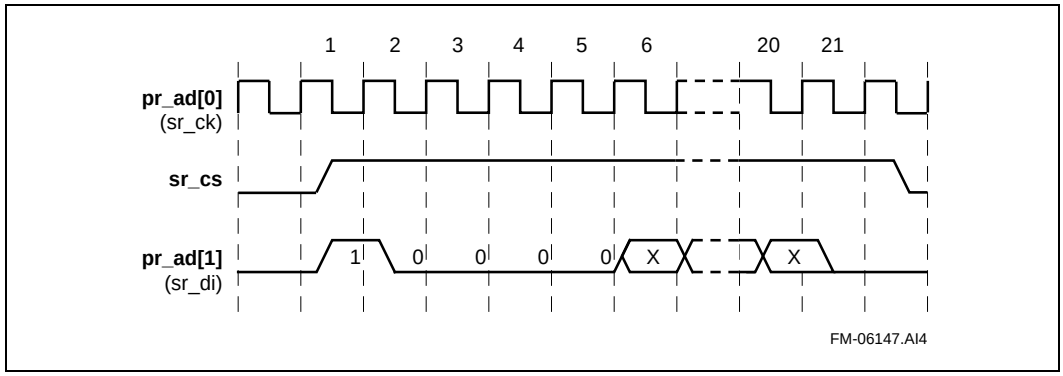


Figure 12-6. Serial ROM Erase Timing Diagram

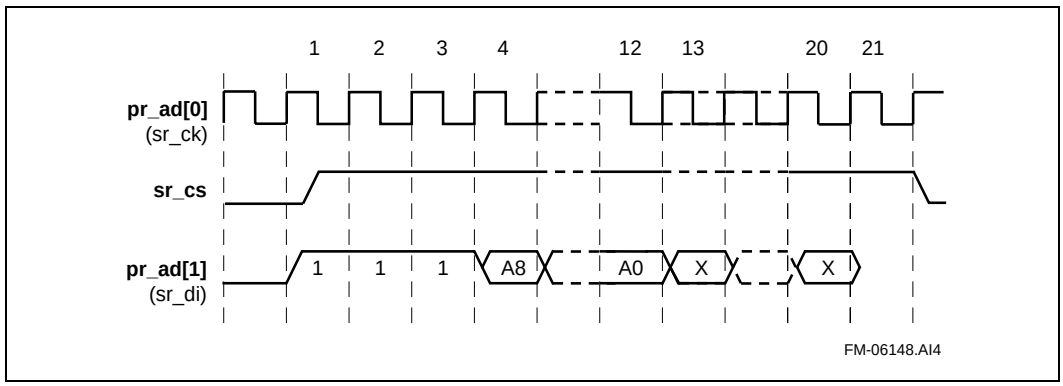


Figure 12-7. Serial ROM Erase All Operation

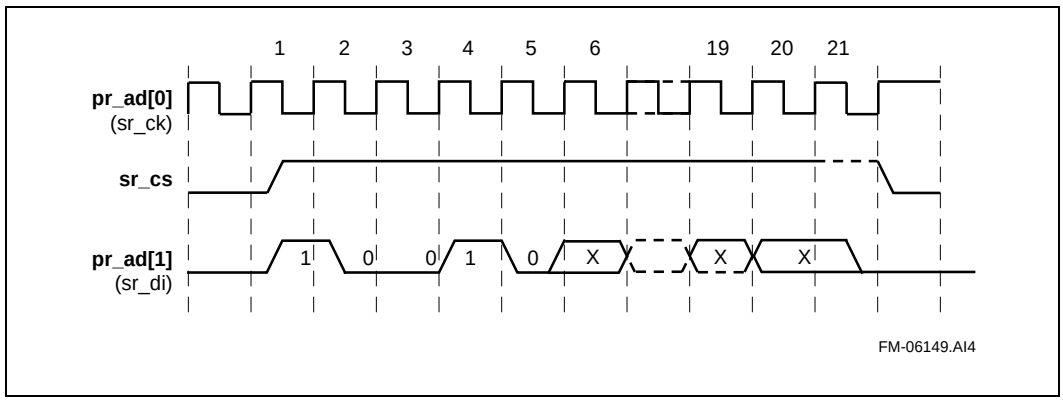
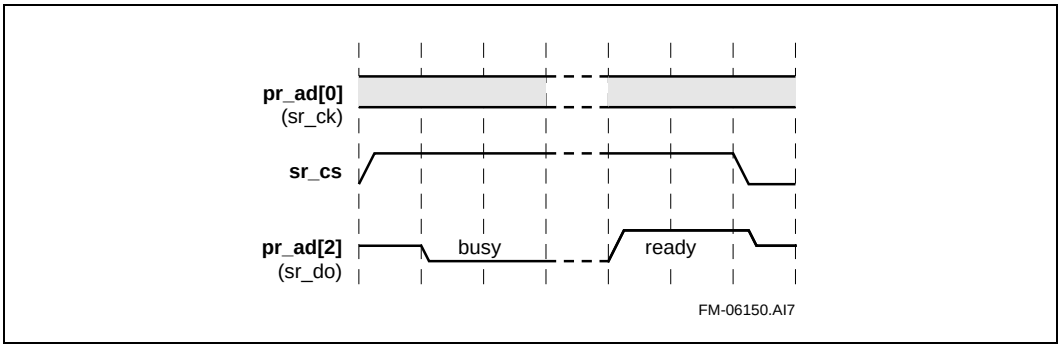


Figure 12-8. Serial ROM Check Status Timing Diagram



Arbitration 13

This chapter describes how the 21554 implements primary and secondary PCI bus arbitration.

13.1 Primary PCI Bus Arbitration

The 21554 implements primary PCI bus request and grant pins, p_req_l and p_gnt_l, that interface to an external primary bus arbiter. These pins are used when the 21554 wants to initiate a transaction on the primary PCI bus.

The 21554 asserts p_req_l whenever a posted write or delayed transaction is queued in upstream buffers. Signal p_req_l remains asserted as long as the posted write and delayed transaction queues contain pending transactions; otherwise, p_req_l is deasserted. However, if the 21554 is keeping p_req_l asserted and the 21554 detects a target retry or target disconnect in response to an ongoing transaction, it deasserts p_req_l one cycle after p_stop_l is detected asserted before re-attempting arbitration for that transaction. p_req_l is deasserted for two clock cycles.

If a prefetchable read is ongoing on the target bus when another delayed read is queued behind it, then the 21554 delays the assertion of p_req_l until it is ensured that there is room in the read data queue for the second delayed read transaction.

If p_gnt_l is asserted when p_req_l is not asserted, then the 21554 parks p_ad, p_cbe_l, and p_par by driving them to valid logic levels. The 64-bit extension signals are not parked. If the primary bus is parked at the 21554 and the 21554 then has a transaction to initiate on the primary bus, it starts the transaction immediately, as long as p_gnt_l was asserted during the previous clock cycle.

13.2 Secondary PCI Bus Arbitration

The 21554 implements an internal secondary PCI bus arbiter supporting nine secondary bus masters, plus the 21554. The internal arbiter may be disabled and an external arbiter used for secondary bus arbitration.

The behavior of the 21554 secondary request is identical to the behavior of the 21554's primary bus request.

13.2.1 Secondary Bus Arbitration Using the Internal Arbiter

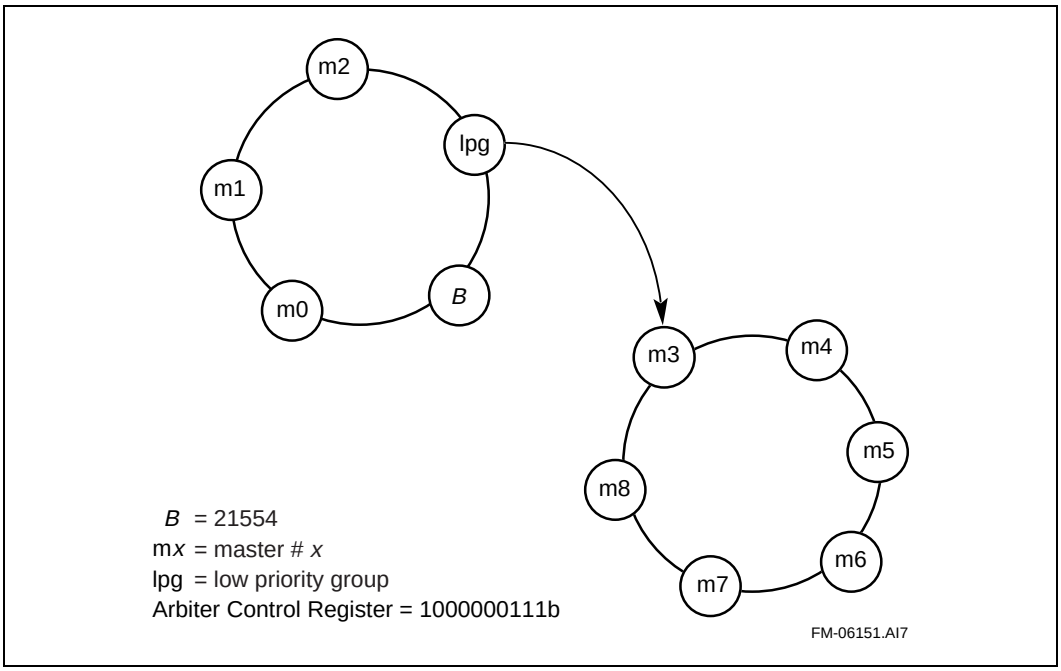
The 21554 enables the secondary bus arbiter when it detects pr_ad[7] high during reset. The 21554 has nine secondary bus request input pins, s_req_l[8:0], and nine secondary bus output grant pins, s_gnt_l[8:0], to support external secondary bus masters. The 21554 secondary bus request and grant signals are connected internally to the arbiter and are not brought out to external pins when the arbiter is enabled.

The secondary arbiter supports a programmable two level rotating priority algorithm. Two groups of masters are assigned, a high priority group and a low priority group. The low priority group as a whole represents one entry in the high priority group. That is, if the high priority group consists of

N masters, then in at least every N+1 transactions the highest priority is assigned to the low priority group. Priority rotates evenly among the low priority group. Therefore, members of the high priority group may be serviced N transactions out of N+1, while one member of the low priority group is serviced once every N+1 transactions. See Figure 13-1 for an example where four masters, including the 21554, are in the high priority group and six masters are in the low priority group. Using this example, if all requests are always asserted, the highest priority would rotate among the masters in the following fashion (high priority members in italics, low priority members in bold):

B, m0, m1, m2, **m3,** *B, m0, m1, m2,* **m4,** *B, m0, m1, m2,* **m5,** *B, m0, m1, m2,* **m6,** *B, m0, m1,* ... etc.

Figure 13-1. Secondary Arbiter Example



Each bus master, including the 21554, may be configured to be in either the low priority group or the high priority group by setting the corresponding priority bit in the Arbiter Control register in device-specific configuration space. If the bit is set to a one, the master is assigned to the high priority group. If the bit is set to a zero, the master is assigned to the low priority group. If all the masters are assigned to one group, then the algorithm defaults to a straight rotating priority among all the masters. After reset, all external masters are assigned to the low priority group and the 21554 is assigned to the high priority group. The 21554 receives highest priority on the target bus every other transaction, and priority rotates evenly among the other masters.

Priorities are reevaluated every time s_frame_l is asserted, at the start of each new transaction on the secondary PCI bus. From this point until the time that the next transaction starts, the arbiter asserts the grant signal corresponding to the highest priority request that is asserted. If a grant for a particular request is asserted, and a higher priority request subsequently asserts, then the arbiter deasserts the asserted grant signal and asserts the grant corresponding to the new higher priority request on the next PCI clock cycle. When priorities are reevaluated, the highest priority is assigned to the next highest priority master relative to the master that initiated the previous transaction. The master that initiated the last transaction has the lowest priority in its group.

If the 21554 detects that a master has failed to assert s_frame_l after 16 cycles of both grant assertion and a secondary idle bus condition, the arbiter deasserts the grant. That master does not receive any more grants until it deasserts its request for at least one PCI clock cycle.

To prevent bus contention, if secondary FRAME# is deasserted the arbiter never asserts one grant signal in the same PCI cycle as it deasserts another. It deasserts one grant, and then asserts the next grant no earlier than one PCI clock cycle later. If s_frame_l is asserted, then the arbiter may deassert one grant and assert another grant during the same PCI clock cycle.

The 21554's internal arbiter parks the secondary PCI bus at the last master to use the bus, that is, an initiator's secondary bus grant remains asserted unless and until another initiator has asserted its secondary bus request. After reset, the internal arbiter parks the secondary bus at the 21554.

13.2.2 Secondary Bus Arbitration Using an External Arbiter

The internal arbiter is disabled when pr_ad[7] is detected low during reset. An external arbiter must then be used. When the internal arbiter is disabled, the 21554 redefines two pins to be external request and grant pins. The s_gnt_l[0] pin is redefined to be the 21554's external request pin, since it is an output. The s_req_l[0] pin is redefined to be the external grant pin, since it is an input. The unused secondary bus grant outputs, s_gnt_l[8:1], are driven high. Unused secondary bus request inputs, s_req_l[8:1], should be pulled high through external resistors.

If s_req_l[0] is asserted and the 21554 has not asserted s_gnt_l[0], then the 21554 parks the s_ad, s_cbe_l, and s_par pins by driving them to valid logic levels. The 64-bit extension signals on the 21554 are not bus parked.

Error Handling 14

This chapter describes how the 21554 handles errors.

14.1 Parity Errors

The 21554 checks, forwards, and generates parity on both the primary and secondary buses. When forwarding transactions, the 21554 forwards the data parity condition as queued, whether it is bad parity or good parity. Table 14-1 gives in detail the 21554's responses to parity errors.

Table 14-1. Parity Error Responses (Sheet 1 of 3)

Type of Error	Type of Transaction	PER[1] P|S	Action Taken
Address Parity Error	Primary Bus Transaction	0 | –	• Responds normally to transaction • Sets primary Detected Parity Error bit • Forwards transaction with correct parity
		1 | –	• Does not respond to transaction • Asserts p_serr_l (if enabled) • Sets primary Detected Parity Error bit
	Secondary Bus Transaction	– | 0	• Responds normally to transaction • Sets secondary Detected Parity Error bit • Forwards transaction with correct parity
		– | 1	• Does not respond to transaction • Asserts s_serr_l (if enabled) • Sets secondary Detected Parity Error bit
Data Parity Error on Primary Bus	Downstream Posted Write	0 | –	• Forwards transaction with parity error • Sets primary Detected Parity Error bit
		1 | –	• Forwards transaction with parity error • Sets primary Detected Parity Error bit • Asserts p_perr_l
	Upstream Posted Write	0 | –	• Transaction completes normally on primary bus
		1 | –	• Transaction completes on primary bus • Sets primary Data Parity Detected bit if p_perr_l is asserted
		1 | 1	• Transaction completes on primary bus • Sets primary Data Parity Detected bit if p_perr_l is asserted • Asserts s_serr_l if no parity error detected on secondary bus

Table 14-1. Parity Error Responses (Sheet 2 of 3)

Type of Error	Type of Transaction	PER[1] P|S	Action Taken
Data Parity Error on Primary Bus	Downstream Delayed Write	0 | –	• Queues and forwards transaction with parity error • Sets primary Parity Error Detected bit
		1 | –	• Returns TRDY# (and STOP# if multiple data phases requested) • Transaction not forwarded • Sets primary Parity Error Detected bit • Asserts p_perr_l
	Upstream Delayed Write	0 | –	• Transaction completes normally on primary bus
		1 | –	• Transaction completes normally on primary bus • Sets primary Data Parity Detected bit if p_perr_l is asserted
		1 | 1	• Transaction completes normally on primary bus • Sets primary Data Parity Detected bit if p_perr_l is asserted • Asserts s_perr_l when returning s_trdy_l to initiator on secondary bus (for both CSR and BAR forwarding mechanisms)
	Downstream Delayed Read	– | –	• The 21554 is returning data, all action is taken by initiator
	Upstream Delayed Read	0 | –	• Returns read data with bad parity to initiator (for both CSR and BAR forwarding mechanisms) • Sets primary Parity Error Detected bit
		1 | –	• Returns read data with bad parity to initiator (for both CSR and BAR forwarding mechanisms) • Sets primary Parity Error Detected bit • Sets primary Data Parity Detected bit • Asserts p_perr_l
	Configuration Register or CSR Write	0 | –	• Writes the data normally • Sets the primary Parity Error Detected bit
		1 | –	• Writes the data normally • Sets the primary Parity Error Detected bit • Asserts p_perr_l
	Configuration register or CSR Read	– | –	• Returns read data normally

Table 14-1. Parity Error Responses (Sheet 3 of 3)

Type of Error	Type of Transaction	PER[1] P|S	Action Taken
Data Parity Error on Secondary Bus	Downstream Posted Write	– | 0	• Transaction completes normally on secondary bus
		– | 1	• Transaction completes on secondary bus • Sets secondary Data Parity Detected bit if s_perr_l is asserted
		1 | 1	• Transaction completes on secondary bus • Sets secondary Data Parity Detected bit if s_perr_l is asserted • Asserts p_serr_l if no parity error detected on primary bus
	Upstream Posted Write	– | 0	• Forwards transaction with parity error • Sets secondary Detected Parity Error bit
		– | 1	• Forwards transaction with parity error • Sets secondary Detected Parity Error bit • Asserts s_perr_l
	Downstream Delayed Write	– | 0	• Transaction completes normally on secondary bus
		– | 1	• Transaction completes normally on secondary bus • Sets secondary Data Parity Detected bit if s_perr_l is asserted
		1 | 1	• Transaction completes normally on secondary bus • Sets secondary Data Parity Detected bit if s_perr_l is asserted • Asserts p_perr_l when returning p_trdy_l to initiator on primary bus (for both CSR and BAR forwarding mechanisms)
	Upstream Delayed Write	– | 0	• Transaction completes normally on secondary bus
		– | 1	• Returns TRDY# (and STOP# if multiple data phases requested) • Transaction not forwarded • Sets secondary Parity Error Detected bit • Asserts s_perr_l
	Downstream Delayed Read	– | 0	• Returns read data with bad parity to initiator (for both CSR and BAR forwarding mechanisms) • Sets secondary Parity Error Detected bit
		– | 1	• Returns read data with bad parity to initiator (for both CSR and BAR forwarding mechanisms) • Sets secondary Parity Error Detected bit • Sets secondary Data Parity Detected bit • Asserts s_perr_l
	Upstream Delayed Read	– | –	• The 21554 is returning data, all action is taken by initiator
	Configuration Register or CSR Write	– | 0	• Writes the data normally • Sets the secondary Parity Error Detected bit
		– | 1	• Writes the data normally • Sets the secondary Parity Error Detected bit • Asserts s_perr_l
	Configuration Register or CSR Read	– | –	• Returns read data normally

1. PER: Parity Error Response bit (Primary | Secondary)

14.2 System Error (SERR#) Reporting

The 21554 has two system error pins. p_serr_l reports system errors on the primary interface, and s_serr_l reports system errors on the secondary interface. For the 21554 to assert the SERR# signal for that interface, the SERR# Enable must be set in the Command configuration register corresponding to that interface. In addition, each device-specific condition has a disable bit for each interface. When a disable bit for a particular condition is set, SERR# assertion is masked for that condition.

SERR# may be asserted for any of the following conditions:

- Address parity error (disabled by the corresponding Parity Error Enable bit)
- Parity error reported on target bus only during a posted write transaction (disabled by the corresponding Parity Error Enable bit)
- Target abort detected during posted write transaction
- Master abort detected during posted write transaction
- Posted write discarded after 2^{24} target retries received from target
- Delayed write request discarded after 2^{24} target retries received from target
- Delayed read request discarded after 2^{24} target retries received from target
- Delayed transaction completion discarded after master timeout counter expired

For the above conditions, SERR# is asserted on the interface corresponding to the location of the initiator of the transaction.

When the SERR# Forward Enable bit in the Chip Control 0 configuration register is set, the 21554 asserts p_serr_l whenever it detects s_serr_l asserted, including those cases where the 21554 has asserted s_serr_l.

Clocking 15

The 21554 supports two clock inputs, p_clk and s_clk. The signal p_clk corresponds to the primary interface and s_clk corresponds to the secondary interface. Both clocks must adhere to the PCI Local Bus specification.

The 21554 may operate in either synchronous or asynchronous mode. The 21554 starts in asynchronous mode during reset, but may switch to synchronous mode after reset if pr_ad[4] is sampled low during reset.

In asynchronous mode, p_clk and s_clk may be asynchronous to each other, that is, they may have any phase relationship and may differ in frequency.

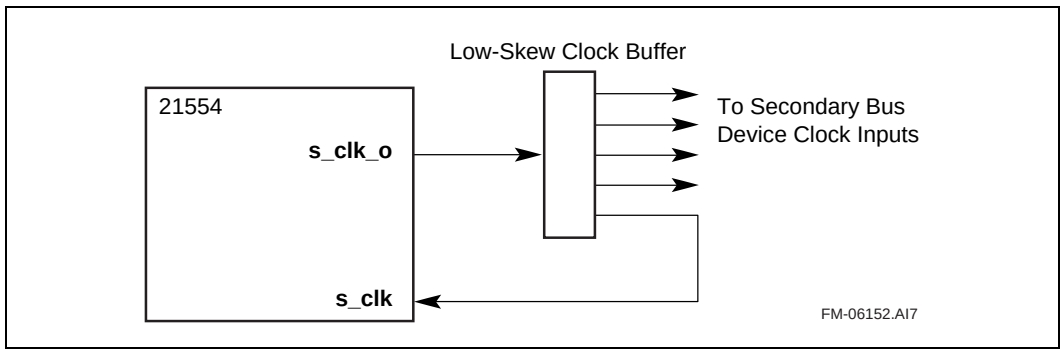
When the 21554 operates in synchronous mode, p_clk and s_clk must operate at the same frequency and have a fixed phase relationship. Operation in synchronous mode saves at least one clock cycle of latency for transactions crossing the bridge.

15.1 21554 Secondary Clock Outputs

If the secondary clock is not supplied independently, the secondary clock output implemented on the 21554 may be used in either synchronous or asynchronous mode. The 21554 secondary clock output, s_clk_o, may be buffered externally for use with secondary bus devices and the 21554 secondary interface clock input, as shown in Figure 15-1. This clock output is a buffered version of p_clk and therefore has the same clock frequency as p_clk. If s_clk_o is used for secondary bus devices, one of the externally buffered clock outputs must be used for the 21554 secondary clock input, s_clk.

Signal s_clk_o is disabled and driven low when the 21554 samples pr_ad[5] low during reset. Signal s_clk_o may also be disabled by setting the s_clk_o Disable bit in the Chip Control 0 configuration register.

Figure 15-1. Synchronous Secondary Clock Generation



Initialization Requirements 16

This chapter summarizes the reset operation and initialization requirements for the 21554.

16.1 Reset Behavior

The 21554 implements a primary reset input, p_rst_l, and a secondary reset output, s_rst_l. The 21554 also implements a Chip Reset bit and a Secondary Reset bit in the Chip Control 0 configuration register.

The device is reset whenever:

- Signal p_rst_l is asserted.
- The Chip Reset bit is written with a 1.
- A power management transition from D3hot to D0 occurs (see Section 16.3).

If the Chip Reset bit is written with a 1, the bit clears itself after chip reset is complete. Chip reset causes all the register values to be reset, and all the queues to be cleared. The primary PCI bus and control signals are tristated as long as either chip reset is occurring or p_rst_l is asserted.

The secondary reset output, s_rst_l, is asserted and remains asserted when any of the following are true:

- The primary reset input, p_rst_l, is asserted.
- The Secondary Reset bit in the 21554 Control register is set to a 1.
- The Chip Reset bit in the 21554 Control register is set to a 1.
- A power management transition from D3hot to D0 occurs (see Section 16.3).

A power management transition from D3hot to D0 or setting the Chip Reset bit causes the Secondary Reset bit to set automatically. When set automatically, the Secondary Reset bit also clears automatically and s_rst_l deasserts after >100 µsec following s_rst_l assertion.

Assertion of s_rst_l by setting the secondary reset bit does not cause the 21554 register state to be reset. However, all the 21554 data buffers are reset. Note that a configuration write is required to clear the secondary reset bit if the bit is set by a configuration write. Therefore, care must be taken if this bit is asserted from the secondary interface.

Table 16-1 summarizes the various 21554 reset mechanisms. Note that s_rst_l is asserted for all reset mechanisms, but how s_rst_l deasserts and whether the device is reset varies from case to case.

Table 16-1. Reset Mechanisms

Reset Mechanism	Reset 21554 Buffers and State?	Assert Secondary Reset Bit?	Deassertion of s_rst_l
p_rst_l	Yes	–	On p_rst_l deassertion
Chip Reset Bit set	Yes	Yes	Automatically after >100 µsec (Secondary Reset bit also clears automatically)
Secondary Reset Bit set	Reset data buffers and primary master state machine	Yes	On clearing of Secondary Reset bit
Transition from $D3_{hot}$ to D0	Yes	Yes	Automatically after >100 µsec (Secondary Reset bit also clears automatically)

16.1.1 Central Function During Reset

The 21554 is selected to be the secondary bus central function when it detects pr_ad[6] low when s_rst_l is asserted. When the 21554 detects this condition, it immediately drives s_ad, s_cbe_l, and s_par low and tristates secondary bus control signals for the duration of secondary bus reset. The 21554 also asserts s_req64_l, but tristates all other 64-bit extension signals.

If pr_ad[6] is detected high during s_rst_l assertion, then another device is acting as a central function on the secondary bus. The 21554 tristates all secondary PCI signals, including s_ad, s_cbe_l, and s_par, for the duration of secondary bus reset. The 21554 does not assert s_req64_l during reset, therefore an external agent must assert s_req64_l to enable the 21554's secondary interface 64-bit extension.

16.2 21554 Initialization

The 21554 supports the following mechanisms for initialization and configuration:

- Preconfiguration using the serial ROM interface (also called serial ROM preload)
- Configuration by the local processor through the secondary interface
- Configuration by the host processor through the primary interface

Initialization may use all of these mechanisms, or only a subset. Initialization must take place:

- After a hardware reset caused by p_rst_l assertion
- After device reset caused by setting the Chip Reset bit in the Reset Control register
- After device reset caused by a power management transition from D3hot to D0.

The 21554 reset consists of the following sequence:

1. Signal p_rst_l asserts or the device is reset. The 21554 tristates all PCI outputs and asserts s_rst_l.
2. Signal p_clk and s_clk start; s_clk_o is a buffered version of p_clk.
3. If pr_ad[6] is low, the 21554 drives s_ad, s_par, and s_cbe_l low for the remainder of s_rst_l assertion, and asserts s_req64_l.
4. Upon deassertion of p_rst_l or on the 1st clock cycle following the completion of chip reset:
   - The value of pr_ad[3] specifies the value of the Primary Lockout configuration bit upon completion of reset.
   - If pr_ad[4] is low, the 21554 switches into synchronous mode.
   - If pr_ad[5] is low, s_clk_o is disabled and driven low.
   - If pr_ad[7] is low, the internal arbiter is disabled.
5. The 21554 deasserts s_rst_l after p_rst_l deassertion, or after 100 µsec following s_rst_l assertion.

16.2.1 Initialization with Serial ROM, Local Processor, and Host Processor

The 21554 initialization using all of the configuration mechanisms uses the following process:

1. Serial Preload	Upon deassertion of p_rst_l or completion of chip reset, the 21554 automatically starts the serial load sequence. The serial load takes approximately 550 serial ROM clock cycles. During this time, the 21554 returns a target retry to any configuration transaction access from either interface. The serial load can overwrite selected PCI read-only registers, program forwarding BAR types and sizes, and configure device-specific configuration registers.
2. Local Processor Initialization	When the serial load is complete, the 21554 configuration registers are now accessible by the local processor from the secondary interface. If the Primary Lockout bit is set, the 21554 continues to return target retry to any configuration accesses from the primary interface (with the exception of the Reset Control configuration register at offset D8h). The local processor may write selected locations that are loadable from the serial ROM, and therefore may be used to change parameters loaded during serial ROM preconfiguration. The local processor may also perform standard PCI configuration of the secondary interface configuration registers. Once the base address registers are mapped and the secondary enables set, the 21554 may accept memory or I/O transactions to its CSR registers. When local processor initialization is complete, the local processor should clear the Primary Lockout bit to allow host initialization. If the Primary Lockout bit is clear after serial preload, the host processor and local processor may access the 21554 concurrently immediately after serial preload is complete. The local processor should *not* change any primary interface preload values that could affect host configuration. This mode of initialization is not recommended unless special care is taken that registers are accessed and initialized in their proper sequence.
3. Host Initialization	After the serial preload and Primary Lockout bit is clear, the host may perform the standard PCI device configuration. Device-specific expansion ROM code may be accessed through the Primary Expansion ROM Base Address register.
4. Normal Operation	

16.2.2 Initialization Without Serial Preload

A serial ROM is supported, but not required, for the 21554 preinitialization. In the case where a serial ROM is not connected to the 21554 or when the first byte read does not contain the preload enable sequence, the 21554 terminates the serial ROM read and configuration space is then available for local processor configuration. The Primary Lockout bit may be set to a one by pulling pr_ad[3] high during chip reset. All primary bus configuration accesses (with the exception of location D8h) then receive target retry until the local processor clears the Primary Lockout bit.

The local processor first must preconfigure registers that would have been preloaded by the serial ROM. This is particularly true of the size and types of the base address registers for forwarding transactions, which upon completion of reset are disabled and request no address space.

Once the local processor preconfigures the necessary registers, normal PCI configuration of the secondary configuration registers can proceed. The local processor then *must* clear the Primary Lockout bit to allow access from the primary bus, unless the Primary Lockout reset value was designated to be low by pulling pr_ad[3] low during reset.

The remainder of the 21554 configuration proceeds as described in Section 16.2.1.

16.2.3 Initialization Without Local Processor

Initialization of the 21554 is possible without a local processor, or without local processor intervention. However, the serial load *must* clear the Primary Lockout bit to allow access of configuration registers from the primary interface. The serial preload must successfully preconfigure the forwarding BAR setup registers, as well as overwrite primary read-only registers as necessary.

Upon completion of the serial preload, all configuration registers are accessible for PCI configuration from the host on the primary bus. The host is then also responsible for configuring the secondary interface and device-specific configuration registers.

16.2.4 Initialization Without Local Processor and Serial Preload

If neither the serial ROM nor a local processor is present, then only the reset values of all the read-only registers are used, and all forwarding BARs are disabled and request no space (since all these registers are set up from the secondary side only). The 21554 configuration registers are accessible, and the 21554 CSR registers can still be mapped into memory or I/O space. However, the Primary Expansion ROM BAR is disabled. A parallel ROM can still be accessed through the CSR mechanism. I/O and configuration transactions can be forwarded through the indirect CSR mechanism, and the I_2O message unit, doorbell registers, and scratchpad registers are all accessible.

16.2.5 Initialization Without Host Processor

Initialization of the 21554 can be performed without a host processor. In this case, the local processor must perform the initialization of the primary configuration registers from the secondary interface.

16.3 Power Management Support

The 21554 implements the PCI Power Management interface on behalf of the subsystem. The 21554 Power Management interface is designed to be flexible in order to meet the varying needs of different types of subsystem functions. In order to fully understand the PCI Power Management interface, please refer to the *PCI Power Management Specification, Rev 1.0.* Some functions may need minimal power management support: the D0 and $D3_{hot}$ power states, without PME# support. Other functions may need all four power states and PME# support. Power management setup is done by serial ROM preload.

The serial ROM preload allows the following power management parameters to be defined:

- Power Management revision number
- D1 power management state support
- D2 power management state support
- PME# support
- Power Management Data register support
- Device Specific Initialization status bit

16.3.1 Transitions Between Power Management States

The 21554 is put into a different power state by writing the Power State bits in the Power Management Control and Status configuration register. Table 16-2 shows the actions that the 21554 takes when transitioning between power states. Although any transition to a lower power state is allowed, all transitions to a higher power state must go to D0.

Table 16-2. Power Management Actions

Original Power State	Next Power State	Action
D0	D1	No action. Subsystem should have been notified by driver.
D0, D1	D2	No action. Subsystem should have been notified by driver.
D0, D1, D2	$D3_{hot}$	No action. Subsystem should have been notified by driver.
Any State	$D3_{cold}$	No action. Powered off.
D1, D2	D0	Set “Transition to D0” status bit and assert s_inta_l if not masked for that event.
$D3_{hot}$	D0	The 21554 performs a chip reset and asserts s_rst_l for 100 µsec. The 21554 performs a serial preload as soon as chip reset is complete.[1]
$D3_{cold}$	D0	Power on. Primary bus reset asserts. No special action needed.

1. To adhere to the $D3_{hot}$ to D0 recovery time stated in the Power Management Specification, the local processor may have to initialize the 21554 and clear the Primary Lockout bit early in the subsystem initialization process.

16.3.2 PME# Support

The 21554 provides optional PME# support. Since the 21554 provides the subsystem Power Management Interface registers, the 21554 must also be the source of the PME# signal for the subsystem. The 21554 implements a primary bus PME# output signal, p_pme_l, that is asserted when the subsystem wants to generate a power management event. The 21554 implements a secondary bus power management input signal, s_pme_l, that the subsystem asserts to notify the 21554 of this power management event.

The 21554 asserts p_pme_l when all of the following are true:

- The 21554 detects s_pme_l asserted low.
- PME# support for the current power state of the 21554 is enabled, as indicated in the Power Management Capabilities register.
- The PME_En bit is set to a 1 in the Power Management Control and Status register.

If the first two conditions have both been met, the 21554 sets the PME Status bit in the Power Management Control and Status register.

Once p_pme_l has been asserted, the 21554 deasserts the signal if either of the following conditions are true:

- The PME Status bit is cleared in the Power Management Control and Status register.
- The PME_En bit is cleared in the Power Management Control and Status register.

The 21554 assumes that s_pme_l is deasserted before the PME_Status bit is cleared in the Power Management Control and Status register. Otherwise, multiple assertions of p_pme_l may occur. If PME# isolation circuitry is required on the primary interface, it must be implemented externally.

16.3.3 Power Management Data Register

The PCI Power Management specification defines an optional data register that can be used for data reporting. A Data Select field in the Power Management Control and Status register selects the type of data to be reported. A Data Scale register provides the scale factor for this data. The 21554 allows implementation of this Data register for static data reporting for the subsystem. The Data Scale value and eight possible data values are loaded into the 21554 through the serial ROM preload operation. The Power Management Data Enable bit in the serial preload sequence enables the use of these data values; otherwise the Data register will read as 0. The value contained in the Data Select field selects which data byte is to be returned when the Power Management Data register is read.

16.4 Compact PCI Hot-Swap Functionality

The 21554 implements hot-swap functionality that allows it to function as a Compact PCI hot-swap controller. The basic components of a Compact PCI Hot- Swap device are:

- Enhanced Capabilities Port (ECP) Compact PCI hot-swap configuration register, at offset ECh.
- Support of the hot-swap event pin, p_enum_l. This signal is routed to the host CPU through the Compact PCI connector. This signal informs the CPU that the configuration of the system has changed; that is, the card has been inserted or is about to be removed.
- Support bi-directional pin, l_stat. This signal is a micro-switch sensor input and a LED control output.

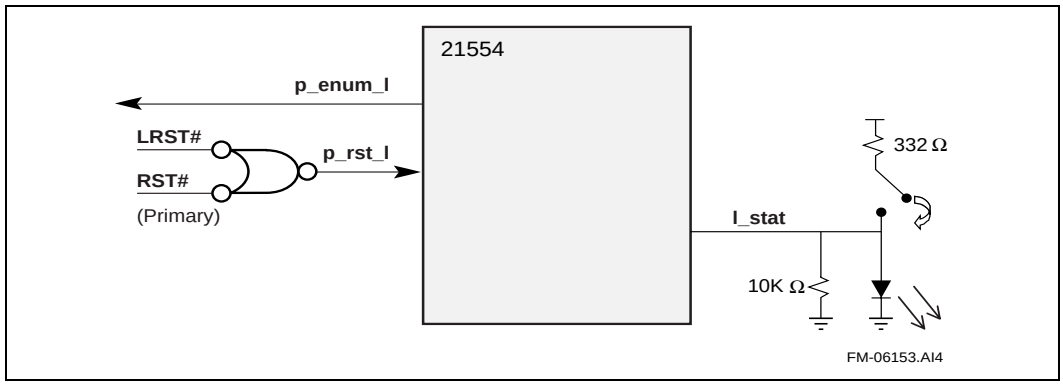
A Compact PCI hot-swap board has a staggered pin arrangement to allow power/ground, signal, and a board inserted indicator to be connected and disconnected in stages. Power and ground are 1st make, last (3rd) break pins. The signal pins are 2nd make, 2nd break pins. The board inserted signal, which is routed to the power conditioning logic, is last (3rd) make, 1st break.

A card ejector handle controls a micro-switch on the card. This micro-switch in turn controls a signal l_stat, which controls the LED and indicates to the 21554 when the ejector is open or closed. A high value on the l_stat input indicates that the ejector handle is open and the LED is turned on. A low value on the l_stat input indicates that the ejector handle is closed (unless overdriven by the 21554) and the LED is off.

The LED may also be controlled independently of the micro-switch and hot-swap functionality by writing the LED On/Off (LOO) Control bit in the Hot-Swap Control register. Figure 16-1 shows how the l_stat signal is used on a Compact PCI hot-swap card.

The 21554 assumes that the card's local reset signal, which is asserted upon card removal or insertion, is OR'ed with the primary bus reset on the card, and then input to the 21554's p_rst_l reset input.

Figure 16-1. Compact PCI Hot-Swap Connections



16.4.1 Add-In Card Insertion

Table 16-3 shows the stages of insertion from the 21554 point of view.

Table 16-3. Hot-Swap Insertion from 21554 Viewpoint

	Event	p_enum_l	p_rst_l	l_stat	s_rst_l	Other
1	1st Make	Z	L	Sampled H[1]	L	Primary PCI tristate.
2	2nd Make	Z	L	Sampled H	L	
3	3rd Make	Z	L	Sampled H	L	
4	LRST# ↑	Z	H	Sampled H	H	Serial preload starts; **l_stat** may be high or low.
5	Ejector Handle Closes	Z	H	Sampled L	H	Ejector handle may also close after serial preload completes.
6	Preload done	L	H	Sampled L	H	Primary Lockout Bit cleared; Set Insertion Status bit.
7	CPU clears Insertion Status	Z	H	Sampled L	H	
8	CPU installs drivers	Z	H	Sampled L	H	

1. The 21554 tristates **l_stat** when sampling this signal.

When an add-in card containing the 21554 is inserted into a powered-on system, power and ground are first supplied to the 21554. The remaining subsystem components do not have power. Assertion of the local reset causes p_rst_l to be asserted low to the 21554. The 21554 tristates all primary bus PCI pins and asserts s_rst_l. Note that if l_stat is asserted while p_rst_l is asserted, the 21554 will not drive s_ad, s_cbe_l, and s_par low, even if it is designated as the secondary bus central function. The 21554 still asserts s_req64_l during reset if designated as the central function.

The Primary PCI signals are contacted. The 21554 performs no additional actions at this stage.

The board is fully inserted, and power is supplied to the rest of the subsystem. The subsystem is kept in reset by the assertion of s_rst_l. When p_rst_l deasserts, the 21554 performs the following actions:

- Deasserts s_rst_l
- Performs the serial preload operation to start the 21554 initialization process
- Deassertion of local reset, and therefore p_rst_l**,** is an application specific function.

Once l_stat is sampled low (ejector handle closed), the serial ROM preload is complete and the Primary Lockout bit is clear, the 21554 sets the insertion status bit in the Hot-Swap Control register and asserts p_enum_l, if enabled to do so. The device-specific Primary Lockout bit enables access to the 21554 configuration registers and is cleared by the local processor or by serial ROM preload.

Host software clears the p_enum_l Insertion Status bit, causing the 21554 to deassert p_enum_l. The host configures the 21554 and the subsystem driver is loaded. The board is now fully initialized. Software may turn on and off the LED by writing the LOO Control bit during or after initialization.

16.4.2 Add-In Card Removal

Table 16-4 shows the stages of removal from the 21554 point of view. Actions taken by the 21554 are highlighted.

Table 16-4. Add-In Card Removal from 21554 Viewpoint

	Event	p_enum_l	p_rst_l	l_stat	s_rst_l	Other
0	Normal Operation	Z	H	Sampled L[1]	H	
1	Ejector open	L	H	Sampled H	H	Set Removal Status bit
2	LS Output Driven	L	H	Driven L	H	
3	CPU Detects ENUM#	L	H	Driven L	H	
4	CPU Clears Removal Status bit	Z	H	Sampled H	H	LED on
5	1st Break LRST# ↓	Z	L	Sampled H	L	Primary PCI tristate
6	2nd Break	Z	L	Sampled H	L	
7	3rd Break	X	X	X	X	Card removed

1. The 21554 tristates **l_stat** when sampling.

The ejector handle to the board is opened. Signal l_stat is asserted high (input to the 21554). If the LOO Control bit is set when the ejector handle opens, the removal condition is not detected until the LOO Control bit is cleared. When the 21554 detects a rising edge on l_stat, it starts the hot-swap removal sequence. The 21554 then drives l_stat low within a couple of p_clk cycles to override the micro-switch state and turn off the LED. The 21554 asserts p_enum_l and sets the Removal Status bit. The LED may be turned on during this period by software setting the LOO Control bit to 1.

Host software brings the card into a quiescent state from a software point of view and clears the Removal Status bit. If the LOO Control bit is 0, the 21554 tristates l_stat and the LED turns on, since the ejector handle is open. If the LOO Control bit is 1, the 21554 continues driving l_stat high, and again the LED is on.

LRST#, and therefore, p_rst_l, asserts, causing the 21554 to tristate the primary interface and assert s_rst_l to the board.

The board is removed.

If l_stat is sampled low after the Removal Status bit has been cleared, then the 21554 assumes that the card has been reinserted. The 21554 then sets the Insertion Status bit and starts the card insertion sequence from that point (step 6 in Table 16-3). The 21554 register state is not cleared and a serial preload operation is not performed.

Diagnostic Mechanisms 17

This chapter describes the implementation of the 21554's JTAG test port according to IEEE Std 1149.1, IEEE Standard Test Access Port and Boundary-Scan Architecture.

The 21554's JTAG test port consists of the following:

- A 5-signal test port interface
- A test access port controller
- An instruction register
- A bypass register
- A boundary-scan register

The JTAG test access port is to be used only while the 21554 is not operating. Table 17-1 lists the JTAG signal pins with a brief description of each pin.

Table 17-1. JTAG Signal Pins

Signal Name	Type	Description
tdi	I	Serial boundary-scan data in
tdo	O	Serial boundary-scan data out
tms	I	JTAG test mode select
tck	I	Boundary-scan clock
trst_l	I	JTAG test access port reset

17.1 Test Access Port Controller

The test access port controller is a finite state machine that interprets IEEE 1149.1 protocols received through the tms signal. The state transitions in the controller are caused by the tms signal on the rising edge of tck. In each state, the controller generates appropriate clock and control signals that control the operation of the test features. After entry into a state, test feature operations are initiated on the rising edge of tck.

17.1.1 Initialization

The test access port controller and the instruction register are initialized when the trst_l input is asserted. The test access port controller enters the test-logic reset state. The instruction register is reset to hold the bypass register instruction. During test-logic reset state, all JTAG logic is disabled, and the device performs normal functions. The test access port controller leaves this state only when an appropriate JTAG test operation sequence is sent on the tms and tck pins.

Note: Signal trst_l must either be asserted or have been asserted after power-up in order for the device to function.

17.2 Instruction Register

The 5-bit instruction register selects the test mode and features. The instruction codes are shown in Table 17-2. These instructions select and control the operation of the boundary-scan and bypass registers. The instruction register is loaded through the tdi pin. The instruction register has a serial shift-in stage from which the instruction is then loaded in parallel.

Table 17-2. JTAG Instruction Register Options

Instruction Register Contents	Instruction Name	Test Register Selected	Operation
0000	EXTEST	Boundary-scan	External test (drives pins from boundary-scan register)
0001	SAMPLE	Boundary-scan	Samples inputs
0010	HIGHZ	Bypass	Tristates all output and I/O pins except the **tdo** pin
0011	CLAMP	Bypass	Drives pins from the boundary-scan register and selects the bypass register for shifts
0100	IDCODE	Idcode	Reads the manufacturer's identification number, the design part number, and the design verification number
0101-1111	BYPASS	Bypass	Selects the bypass register for shifts

17.3 Bypass Register

The bypass register is a 1-bit shift register that provides a means for effectively bypassing the JTAG test logic through a single-bit serial connection through the device from tdi to tdo. At board-level testing, this helps reduce the overall length of the scan ring.

17.4 Boundary-Scan Register

The boundary-scan register (BSR) is a single-shift register-based path formed by boundary-scan cells placed at the device's signal pins. The register is accessed through the JTAG port's tdi and tdo pins.

Each boundary-scan cell operates in conjunction with the current instruction and the current state in the test access port controller state machine. The function of the BSR cells is determined by the associated pins, as follows:

- Input-only pins: the boundary-scan cell is basically a 1-bit shift register. The cell supports sample and shift operations.
- Output-only pins: the boundary-scan cell comprises a 1-bit shift register and an output multiplexer. The cell supports the sample, shift, and drive output functions.
- Bidirectional pins: the boundary-scan cell is identical to the output-only pin cell, but it captures test data from the incoming data line. The cell supports sample, shift, drive output, and hold output functions.

VPD Support 18

The 21554 provides Vital Product Data (VPD) support through its serial ROM interface. Note that VPD support in the Expansion ROM as described in the *PCI Local Bus Specification, Revision 2.1*, is transparent to the 21554 and is not described in this document.

VPD is stored in the last 3K bits (384 bytes) of the serial ROM. The first 1K bits (128 bytes) of the VPD space are designated as read only and cannot be written from the VPD serial ROM register interface. The upper 2K bits (256 bytes) are designated as read/write from the VPD serial ROM register interface. Only VPD data can be accessed through this interface. To read or write any serial ROM location, the CSR serial ROM access mechanism described in Chapter 12 should be used.

18.1 Reading VPD Information

A read can occur to any location in VPD space. Valid VPD byte addresses are 17F:000h. To read VPD information from the serial ROM, the following steps must be taken:

1. The VPD address and VPD Flag bits are written. This requires a write to bytes E7:E6h, where the low 9 bits carry the VPD byte address and bit 15 is a 0, indicating a read operation. The 21554 adds the VPD base address, 080h, to the VPD byte address to obtain the serial ROM address and perform a read of 4 bytes. The VPD address has no alignment requirements – it may start on any byte boundary.
2. The VPD Flag bit is polled. When the 21554 returns a 1, the read is complete.
3. The VPD information may be read from the VPD Data register at bytes EB:E8. Byte 0 contains the data at the location referenced by the VPD Address register. Bytes 3:1 contain successive bytes.

The 21554 always performs a 4-byte read, regardless of the VPD byte address. Therefore, if the VPD byte address is one of the last 3 bytes in VPD space, the serial ROM address wraps. The remaining 1, 2, or 3 bytes contain invalid data and should be ignored.

The VPD Address and VPD Data registers should not be written while a VPD read operation is taking place, otherwise results are unpredictable.

18.2 Writing VPD Information

A write can occur only to the last 2 Kb (256 bytes) of VPD Space. Valid VPD byte addresses for write operations are 17F:080h. To write VPD information from the serial ROM, the following steps must be taken:

1. The VPD data register is written with 4 bytes of data. Byte 0 contains the data to be written to the location referenced by the VPD byte address. The value in bytes 3:1 of the VPD Data register is written to successive byte locations in VPD space.
2. The VPD address and VPD Flag bits are written. This requires a write to bytes E7:E6h, where the low 9 bits carry the VPD byte address and bit 15 is a 1, indicating a write operation. The 21554 adds the VPD base address, 080h, to the VPD byte address to obtain the serial ROM address and perform a write of 4 bytes. The VPD address has no alignment requirements – it may start on any byte boundary.
3. The VPD Flag bit is polled. When the 21554 returns a 0, the write is complete.

If a write is attempted to a location in the first 1Kb of serial ROM space (address bits 8:7 is 00b), then the 21554 does not perform the write operation and clears the flag bit immediately. If the VPD byte address is one of the last three byte locations in VPD space, then the 21554 only completes those writes to the end of VPD space, that is, it performs either a 3-, 2-, or 1-byte write.

The 21554 tracks whether the serial ROM is enabled for writes. If the serial ROM is write disabled when a VPD write is attempted, the 21554 first automatically performs a write enable operation to the serial ROM. The 21554 does not automatically perform this operation for serial writes using the CSR mechanism.

The VPD Address and VPD Data registers should not be written while a VPD write operation is taking place, otherwise results are unpredictable.

Special Applications 19

This chapter describes Primary and Secondary Bus VGA support.

19.1 Primary Bus VGA Support

The 21554 provides hardware support that allows configuration of itself as a VGA device. The primary class code should be preloaded through the serial ROM or loaded by the local processor with the value for a VGA device (Base Class 03h, Sub-Class 00h, Programming Interface 00h). This allows the 21554 to present itself to the host as a VGA device. The VGA Enable field in the Chip Control 0 configuration register should be set to 01b to enable decoding of VGA transactions on the primary bus for forwarding to the secondary bus. These bits can be set through serial ROM preload, or from a primary or secondary bus configuration write. The addresses that are decoded are:

Memory addresses 000BFFFFh : 000A0000h

I/O addresses:

AD[9:0]	3BBh:3B0h 3DFh:3C0h
AD[31:16]	000h
AD[15:10]	are not decoded

No address translation is performed on these addresses.

The 21554 cannot be enabled as a snooping agent on the primary bus. This is because the 21554 cannot guarantee that it can buffer and forward all palette writes, since the 21554 has finite buffer space and no backoff mechanism when snooping. The 21554 should not be configured to appear as a VGA device in those applications where it may try to configure the 21554 as a snooping agent.

The parallel ROM may be used to store VGA BIOS code, which is mapped through the Primary Expansion ROM Base Address register.

19.2 Secondary Bus VGA Support

The 21554 can be enabled to decode VGA transactions on the secondary bus for forwarding to the primary bus. This is done by setting the VGA Enable field in the Chip Control 0 configuration register to 10b. The addresses that are decoded are the same as for the primary VGA decode, and again the addresses are not translated.

Upstream forwarding of VGA transactions may be useful for applications that want to allow access to a primary bus VGA device frame buffer by local processors in intelligent I/O or embedded subsystems.

Note: VGA decoding must *not* be enabled for both the primary and secondary interface. The value 11b is illegal for the VGA Enable field and can yield unpredictable results.

Serial ROM and Register Reset Summary 20

This chapter contains a summary of the serial ROM preload sequence and register reset and access.

20.1 Serial ROM Preload Sequence

Table 20-1 shows the serial preload sequence for the 21554 registers. Note that not all bits in the sequence are actually used. Bits that are not used must be 0.

Table 20-1. Serial Preload Sequence (Sheet 1 of 3)

Byte offset	Description
00h	[7:6] 10b to enable serial preload [5:0] 000000b (Reserved)
01h	00000000b (Reserved)
02h	00000000b (Reserved)
03h	00000000b (Reserved)
04h	Primary Programming Interface
05h	Primary Sub-Class Code
06h	Primary Base Class Code
07h	Subsystem Vendor ID [7:0]
08h	Subsystem Vendor ID [15:8]
09h	Subsystem ID [7:0]
0Ah	Subsystem ID [15:8]
0Bh	Primary Minimum Grant
0Ch	Primary Maximum Latency
0Dh	Secondary Programming Interface
0Eh	Secondary Sub-Class Code
0Fh	Secondary Base Class Code
10h	Secondary Minimum Grant
11h	Secondary Maximum Latency
12h	Downstream Memory 0 Setup [7:0] Bits [0, 7:4] are not loaded and should be 0.
13h	Downstream Memory 0 Setup [15:8] Bits [11:8] are not loaded and should be 0.
14h	Downstream Memory 0 Setup [23:16]
15h	Downstream Memory 0 Setup [31:24]

Table 20-1. Serial Preload Sequence (Sheet 2 of 3)

Byte offset	Description
16h	Downstream I/O or Memory 1 Setup [7:0] Bits [5:4] are not loaded and should be 0.
17h	Downstream I/O or Memory 1 Setup [15:8]
18h	Downstream I/O or Memory 1 Setup [23:16]
19h	Downstream I/O or Memory 1 Setup [31:24]
1Ah	Downstream Memory 2 Setup [7:0] Bits [0, 7:4] are not loaded and should be 0.
1Bh	Downstream Memory 2 Setup [15:8] Bits [11:8] are not loaded and should be 0.
1Ch	Downstream Memory 2 Setup [23:16]
1Dh	Downstream Memory 2 Setup [31:24]
1Eh	Downstream Memory 3 Setup [7:0] Bits [0, 7:4] are not loaded and should be 0.
1Fh	Downstream Memory 3 Setup [15:8] Bits [11:8] are not loaded and should be 0.
20h	Downstream Memory 3 Setup [23:16]
21h	Downstream Memory 3 Setup [31:24]
22h	Downstream Memory 3 Setup Upper 32 Bits [7:0]
23h	Downstream Memory 3 Setup Upper 32 Bits [15:8]
24h	Downstream Memory 3 Setup Upper 32 Bits [23:16]
25h	Downstream Memory 3 Setup Upper 32 Bits [31:24]
26h	Bit [0]: Primary Expansion ROM Setup [24] (enable) Bits [3:1]: Not loaded. Should be 0. Bits [7:4]: Primary Expansion ROM Setup [15:11]
27h	Primary Expansion ROM Setup [23:16]
28h	Upstream I/O or Memory 0 Setup [7:0] Bits [5:4] are not loaded and should be 0.
29h	Upstream I/O or Memory 0 Setup [15:8]
2Ah	Upstream I/O or Memory 0 Setup [23:16]
2Bh	Upstream I/O or Memory 0 Setup [31:24]
2Ch	Upstream Memory 1 Setup [7:0] Bits [0, 7:4] are not loaded and should be 0.
2Dh	Upstream Memory 1 Setup [15:8] Bits [11:8] are not loaded and should be 0.
2Eh	Upstream Memory 1 Setup [23:16]
2Fh	Upstream Memory 1 Setup [31:24]
30h	Chip Control 0 [7:0]
31h	Chip Control 0 [15:8] Bits [13:12] are not loaded and should be 0.
32h	Chip Control 1 [7:0]

Table 20-1. Serial Preload Sequence (Sheet 3 of 3)

Byte offset	Description
33h	Chip Control 1 [15:8]
34h	Arbiter Control [7:0]
35h	Arbiter Control [15:7] Bits [15:10] are not loaded and should be 0.
36h	Primary SERR# Disable Bit [7] is not loaded and should be 0.
37h	Secondary SERR# Disable Bit [7] is not loaded and should be 0.
38h	Power Management Data 0
39h	Power Management Data 1
3Ah	Power Management Data 2
3Bh	Power Management Data 3
3Ch	Power Management Data 4
3Dh	Power Management Data 5
3Eh	Power Management Data 6
3Fh	Power Management Data 7
40h	Reserved
41h	[1:0] 00b (Reserved) [2] BiST Supported [3] Power Management Data Register Enable [5:4] Power Management Control and Status [14:13] [7:6] Power Management Capabilities Register [1:0]
42h	[0] Power Management Capabilities Register [2] [1] Power Management Capabilities Register [5] [7:2] Power Management Capabilities Register [14:9]

20.2 Register Reset and Access Summary

This section summarizes the 21554 configuration and CSR registers, along with their reset values and access types. Some notes for interpreting Table 20-2 follow:

- Byte offsets that are specific to the primary or secondary interfaces are followed by a (P) or (S) respectively. All other byte offsets refer to both the primary and secondary address spaces. Configuration registers are listed in order of primary byte offset.
- For read and write access, the following apply:
  - **Y**: accessible from both primary and secondary interface
  - **N**: not accessible from either interface
  - **Primary**: for writes, only write from primary interface; for reads, reads as 0 from secondary interface
  - **Secondary**: for writes, only write from secondary interface; for reads, reads as 0 from primary interface
  - Special cases such as W1TC, W1TS, and R0TS are noted.
- Some registers contain fields or bits with different access types such as R, R/W, and W1TC, which are not detailed in this table. See the individual register description for information of this granularity.
- Not all bits in every register denoted as preloadable are necessarily preloaded. See the individual register description for information of this granularity.
- "Via Setup" refers to the Base Address Setup register corresponding to that base address register.

Table 20-2. Configuration Register Summary (Sheet 1 of 4)

Byte Offset	Register Name	Reset Value	Preload	Write Access	Read Access
01:00h 41:40h	Vendor ID	1011		N	Y
03:02h 43:42h	Device ID	0046h		N	Y
05:04h (P) 45:44h (S)	Primary Command	0000h		Y	Y
07:06h (P) 47:46h (S)	Primary Status	0290h		Y	Y
08h 48h	Revision ID	Device-dependent		N	Y
0B:09h (P) 4B:49h (S)	Primary Class Code	068000h	Y	Secondary	Y
0Ch (P) 4Ch (S)	Primary Cache Line Size	00h		Y	Y
0Dh (P) 4Dh (S)	Primary MLT	00h		Y	Y
0Eh 4Eh	Header Type	00h		N	Y
0Fh 4Fh	BiST	00h	Y	[6] Y [7,3:0] Sec.	Y

Table 20-2. Configuration Register Summary (Sheet 2 of 4)

Byte Offset	Register Name	Reset Value	Preload	Write Access	Read Access
13:10h (P) 53:50h (S)	Primary CSR and Memory 0 BAR	00000000h	Via Setup	Y	Y
17:14h (P) 57:54h (S)	Primary CSR I/O BAR	00000001h		Y	Y
1B:18h (P) 5B:58h (S)	Downstream Memory 1 BAR	00000000h	Via Setup	Via Setup	Y
1F:1Ch (P) 5F:5Ch (S)	Downstream Memory 2 BAR	00000000h	Via Setup	Via Setup	Y
23:20h (P) 63:60h (S)	Downstream Memory 3 BAR	00000000h	Via Setup	Via Setup	Y
27:24h (P) 67:64h (S)	Downstream Memory 3 Upper 32 Bits	00000000h	Via Setup	Via Setup	Y
2B:28h 6B:67h	Reserved	00000000h		N	Y
2D:2Ch 6D:6Ch	Subsystem Vendor ID	0000h	Y	Secondary	Y
2F:2Eh 6F:6Eh	Subsystem ID	0000h	Y	Secondary	Y
33:30h (P) 73:70h (S)	Primary Expansion ROM BAR	00000000h	Via Setup	Via Setup	Y
34h 74h	Capabilities Pointer	DCh		N	Y
37:35h 77:75h	Reserved	000000h		N	Y
3B:38h 7B:78h	Reserved	00000000h		N	Y
3Ch (P) 7Ch (S)	Primary Interrupt Line	00h		Y	Y
3Dh (P) 7Dh (S)	Primary Interrupt Pin	01h		N	Y
3Eh (P) 7Eh (S)	Primary MIN_GNT	00h	Y	Secondary	Y
3Fh (P) 7Fh(S)	Primary MAX_LAT	00h	Y	Secondary	Y
45:44h (P) 05:04h (S)	Secondary Command	0000h		Y	Y
47:46h (P) 07:06h (S)	Secondary Status	0290h		Y	Y
4B:49h (P) 0B:09h (S)	Secondary Class Code	068000h	Y	N	Y
4Ch (P) 0Ch (S)	Secondary Cache Line Size	00h		Y	Y
4Dh (P) 0Dh (S)	Secondary MLT	00h		Y	Y

Table 20-2. Configuration Register Summary (Sheet 3 of 4)

Byte Offset	Register Name	Reset Value	Preload	Write Access	Read Access
53:50h (P) 13:10h (S)	Secondary CSR Memory BAR	00000000h		Y	Y
57:54h (P) 17:14h (S)	Secondary CSR I/O BAR	00000001h		Y	Y
5B:58h (P) 1B:18h (S)	Upstream I/O or Memory 0 BAR	00000000h	Via Setup	Via Setup	Y
5F:5Ch (P) 1F:1Ch (S)	Upstream Memory 1 BAR	00000000h	Via Setup	Via Setup	Y
63:60h (P) 23:20h (S)	Upstream Memory 2 BAR	00000000h	Via Chip Control 1	Via Chip Control 1	Y
67:64h (P) 27:24h (S)	Reserved	00000000h		N	Y
73:70h (P) 33:30h (S)	Reserved	00000000h		N	Y
7Ch (P) 3Ch (S)	Secondary Interrupt Line	00h		Y	Y
7Dh (P) 3Dh (S)	Secondary Interrupt Pin	00h		N	Y
7Eh (P) 3Eh (S)	Secondary MIN_GNT	00h	Y	N	Y
7Fh (P) 3Fh (S)	Secondary MAX_LAT	00h	Y	N	Y
83:80h	Downstream Configuration Address	Indeterminate		Primary	Y
87:84h	Downstream Configuration Data	Indeterminate		Primary	Primary
8B:88h	Upstream Configuration Address	Indeterminate		Secondary	Y
8F:8Ch	Upstream Configuration Data	Indeterminate		Secondary	Secondary
90h	Configuration Own Bits byte 0	00h		N	Primary Read-0-to-Set
91h	Configuration Own Bits byte 1	00h		N	Secondary Read-0-to-Set
93:92h	Configuration CSR	0000h		Y	Y
97:94h	Downstream Memory 0 Translated Base	Indeterminate		Y	Y
9B:98h	Downstream I/O or Memory Translated Base	Indeterminate		Y	Y
9F:9Ch	Downstream Memory 2 Translated Base	Indeterminate		Y	Y
A3:A0h	Downstream Memory 3 Translated Base	Indeterminate		Y	Y
A7:A4h	Upstream I/O or Memory 0 Translated Base	Indeterminate		Y	Y
AB:A8h	Upstream Memory 1 Translated Base	Indeterminate		Y	Y

Table 20-2. Configuration Register Summary (Sheet 4 of 4)

Byte Offset	Register Name	Reset Value	Preload	Write Access	Read Access
AF:ACh	Downstream Memory 0 Setup	FFFFF000h	Y	Secondary	Y
B3:B0h	Downstream I/O or Memory 1 Setup	00000000h	Y	Secondary	Y
B7:B4h	Downstream Memory 2 Setup	00000000h	Y	Secondary	Y
BB:B8h	Downstream Memory 3 Setup	00000000h	Y	Secondary	Y
BF:BCh	Upper 32 Bits Downstream Memory 3 Setup	00000000h	Y	Secondary	Y
C3:C0h	Primary Expansion ROM Setup	00000000h	Y	Secondary	Y
C7:C4h	Upstream I/O or Memory 0 Setup	00000000h	Y	Secondary	Y
CB:C8h	Upstream Memory 1 Setup	00000000h	Y	Secondary	Y
CD:CCh	Chip Control 0	0y00h y=0h or 4h (strapped)	Y	[15:11,9:0] Y [10] Sec.	Y
CF:CEh	Chip Control 1	0000h	Y	Y	Y
D1:D0h	Chip Status	0000h		W1TC	Y
D3:D2h	Arbiter Control	0200h	Y	Y	Y
D4h	Primary SERR# Disable	00h	Y	Y	Y
D5h	Secondary SERR# Disable	00h	Y	Y	Y
D7:D6h	Reserved	0000h		N	Y
DF:D8h	Reset Control	0000h		Primary	Y
DCh	PM Cap ID	01h		N	Y
DDh	PM Next Ptr	E4h		N	N
DF:DEh	PM Capabilities	0001b	Y	Secondary	Y
E1:E0h	PM Control and Status	0000h	Y	Y	Y
E2h	PM Bridge Support Extensions	00h		N	Y
E3h	PM Data	00h	Y	N	Y
E4h	VPD Cap ID	03h		N	Y
E5h	VPD Next Ptr	ECh		N	N
E7:E6h	VPD Address	Indeterminate		Y	Y
EB:E8h	VPD Data	Indeterminate		Y	Y
ECh	HS Cap ID	06h		N	N
EDh	HS Next Ptr	00h		N	N
EEh	HS Control	00x1000b		Y	N
FF:F0h	Reserved	00000000h		N	Y

20.3 CSR Register Summary

The CSR registers are memory-mapped in the Primary CSR and Memory 0 Base Address window and the Secondary CSR Base Address window. They are I/O-mapped in the Primary CSR I/O Base Address window and the Secondary CSR I/O Base Address window. Offsets are referenced from these base addresses. Table 20-3 shows a summary of the CSR registers.

Table 20-3. CSR Summary (Sheet 1 of 3)

Byte Offset	Register Name	Reset Value	Write Access	Read Access
003:000h	Downstream Configuration Address	Indeterminate	Primary	Y
007:004h	Downstream Configuration Data	Indeterminate	Primary	Primary
00B:008h	Upstream Configuration Address	Indeterminate	Secondary	Y
00F:00Ch	Upstream Configuration Data	Indeterminate	Secondary	Secondary
010h	Configuration Own byte 0	00h	N	Primary Read-0-to-set
011h	Configuration Own byte 1	00h	N	Secondary Read-0-to-set
013:012h	Configuration CSR	0000h	Y	Y
017:014h	Downstream I/O Address	Indeterminate	Primary	Y
01B:018h	Downstream I/O Data	Indeterminate	Primary	Primary
01F:01Ch	Upstream I/O Address	Indeterminate	Secondary	Y
023:020h	Upstream I/O Data	Indeterminate	Secondary	Secondary
024h	I/O Own byte 0	00h	N	Primary Read-0-to-set
025h	I/O Own byte 1	00h	N	Secondary Read-0-to-set
027:026h	I/O Own Control and Status	0000h	Y	Y
028h	Lookup Table Offset	Indeterminate	Y	Y
02B:029h	Reserved	000h	N	Y
02F:02Ch	Lookup Table Data	Indeterminate	Y	Y
033:030h	I_2O Outbound Post Status	00000000h	N	Y
037:034h	I_2O Outbound Post Mask	00000004h	Y	Y
03B:038h	I_2O Inbound Post Status	00000000h	N	Y
03F:03Ch	I_2O Inbound Post Mask	00000004h	Y	Y
043:040h	I_2O Inbound Queue	Indeterminate	Primary	Primary
047:044h	I_2O Outbound Queue	Indeterminate	Primary	Primary
04B:048h	I_2O Inbound Free Head Pointer	Indeterminate	Y	Y
04F:04Ch	I_2O Inbound Post Tail Pointer	Indeterminate	Y	Y
053:050h	I_2O Outbound Free Tail Pointer	Indeterminate	Y	Y

Table 20-3. CSR Summary (Sheet 2 of 3)

Byte Offset	Register Name	Reset Value	Write Access	Read Access
057:054h	I_2O Outbound Post Head Pointer	Indeterminate	Y	Y
05B:058h	I_2O Inbound Post Counter	0000000h	Secondary	Y
05F:05Ch	I_2O Inbound Free Counter	0000000h	Secondary	Y
063:060h	I_2O Outbound Post Counter	0000000h	Secondary	Y
067:064h	I_2O Outbound Free Counter	0000000h	Secondary	Y
06B:068h	Downstream Memory 0 Translated Base	Indeterminate	Y	Y
06F:06Ch	Downstream I/O or Memory 1 Translated Base	Indeterminate	Y	Y
073:070h	Downstream Memory 2 Translated Base	Indeterminate	Y	Y
077:074h	Downstream Memory 3 Translated Base	Indeterminate	Y	Y
07B:078h	Upstream I/O or Memory 0 Translated Base	Indeterminate	Y	Y
07F:07Ch	Upstream Memory 1 Translated Base	Indeterminate	Y	Y
081:080h	Chip Control CSR	0000h	Y	Y
083:082h	Chip Status CSR	0000h	W1TC	Y
085:084h	Chip Set IRQ Mask	FFFFh	W1TS	Y
087:086	Chip Clear IRQ Mask	FFFFh	W1TC	Y
08B:088h	Upstream Page Boundary IRQ 0	00000000h	W1TC	Y
08F:08Ch	Upstream Page Boundary IRQ 1	00000000h	W1TC	Y
093:090h	Upstream Page Boundary IRQ Mask 0	FFFFFFFFh	Y	Y
097:094h	Upstream Page Boundary IRQ Mask 1	FFFFFFFFh	Y	Y
099:098h	Primary Clear IRQ	0000h	W1TC	Y
09B:09Ah	Secondary Clear IRQ	0000h	W1TC	Y
09D:09Ch	Primary Set IRQ	0000h	W1TS	Y
09F:09Eh	Secondary Clear IRQ	0000h	W1TS	Y
0A1:0A0h	Primary Clear IRQ Mask	FFFFh	W1TC	Y
0A3:0A2h	Secondary Clear IRQ Mask	FFFFh	W1TC	Y
0A5:0A4h	Primary Set IRQ Mask	FFFFh	W1TS	Y
0A7:0A6h	Secondary Set IRQ Mask	FFFFh	W1TS	Y
0AB:0A8h	Scratchpad 0	Indeterminate	Y	Y
0AF:0ACh	Scratchpad 1	Indeterminate	Y	Y
0B3:0B0h	Scratchpad 2	Indeterminate	Y	Y
0B7:0B4h	Scratchpad 3	Indeterminate	Y	Y
0BB:0B8h	Scratchpad 4	Indeterminate	Y	Y
0BF:0BCh	Scratchpad 5	Indeterminate	Y	Y
0C3:0C0h	Scratchpad 6	Indeterminate	Y	Y
0C7:0C4h	Scratchpad 7	Indeterminate	Y	Y
0C9:0C8h	ROM Setup	7E00h	Y	Y

Table 20-3. CSR Summary (Sheet 3 of 3)

Byte Offset	Register Name	Reset Value	Write Access	Read Access
0CAh	ROM Data	Indeterminate	Y	Y
0CBh	Reserved	00h	N	Y
0CE:0CCh	ROM Address	000400h	Y	Y
0CFh	ROM Control	0000h	Y	Y
0FF:0D0h	Reserved	00000000h	N	Y
1FFh:100h	Upstream Memory 2 Lookup Table	Indeterminate	Y	Y
FFF:200h	Reserved	00000000h	N	Y

Support, Products, and Documentation

If you need technical support, a *Product Catalog*, or help deciding which documentation best meets your needs, visit the Intel World Wide Web Internet site:

http://www.intel.com

Copies of documents that have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling **1-800-332-2717** or by visiting Intel's website for developers at:

http://developer.intel.com

You can also contact the Intel Massachusetts Information Line or the Intel Massachusetts Customer Technology Center. Please use the following information lines for support:

For documentation and general information:	
Intel Massachusetts Information Line	
United States:	1–800–332–2717
Outside United States:	1–303-675-2148
Electronic mail address:	techdoc@intel.com

For technical support:	
Intel Massachusetts Customer Technology Center	
Phone (U.S. and international):	1–978–568–7474
Fax:	1–978–568–6698
Electronic mail address:	techsup@intel.com